A FINANCIAL PLANNING DECISION SUPPORT SYSTEM

FOR SMALL MEDIUM ENTERPRISE

A thesis submitted to the Graduate School in partial fulfillment of the requirements for the degree of Masters of Science (Information Technology)

Universiti Utara Malaysia

by

Pong Chia Lee

Sekolah Siswazah
(Graduate School)
Universiti Utara Malaysia

PERAKUAN KERJA KERTAS PROJEK
(Certification of Project Paper)

Saya, yang bertandatangan, memperakukan bahawa
(I, the undersigned, certify that)

PONG CHIA LEE

calon untuk Ijazah
(candidate for the degree of) Sarjana Sains (Teknologi Maklumat)

telah mengemukakan kertas projek yang bertajuk
(has presented his/ her project paper of the following title)

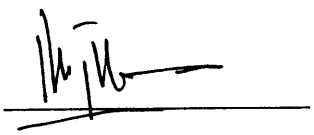
A FINANCIAL PLANNING DECISION SUPPORT SYSTEM

FOR SMALL MEDIUM ENTERPRISE

seperti yang tercatat di muka surat tajuk dan kulit kertas projek
(as it appears on the title page and front cover of project paper)

bahawa kertas projek tersebut boleh diterima dari segi bentuk serta kandungan, dan meliputi bidang ilmu dengan memuaskan.
(that the project paper acceptable in form and content, and that a satisfactory knowledge of the field is covered by the project paper).

Nama Penyelia
(Name of Supervisor) : Prof. Madya Nazib Nordin

Tandatangan
(Signature) : 

Tarikh
(Date) : 1st November 2000

PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a post graduate degree from the Universiti Utara Malaysia, I agree that the Universiti Library may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by my supervisor(s) or, in their absence, by the Dean of the Graduate School. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to Universiti Utara Malaysia for any scholarly use which may be made of any material from my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or in part should be addressed to:

Dean of Graduate School
Universiti Utara Malaysia
06010 UUM Sintok
Kedah Darul Aman

ABSTRACT (BAHASA MALAYSIA)

Projek ini bertujuan untuk membangunkan satu sistem sokongan pemutusan (DSS) perancangan kewangan kos rendah dan berorientasikan pengguna akhir untuk perusahaan kecil dan sederhana di Malaysia. Sistem sokongan pemutusan (DSS) kos rendah dan ramah pengguna adalah penting untuk membantu pembuat-pembuat keputusan bagi perusahaan kecil dan sederhana, dalam melaksanakan perancangan kewangan masa depan memandangkan sistem sokongan pemutusan sedia ada adalah mahal dan tidak serasi untuk kegunaan perusahaan kecil dan sederhana di Malaysia. Integrasi Microsoft Excel 2000 dan Microsoft Visual Basic 6.0 telah digunakan untuk membangunkan sistem ini. Model-model kewangan sistem ini telah dibentuk dengan menggunakan Microsoft Excel 2000. Antaramuka-antaramuka sistem ini pula telah dibangunkan dengan menggunakan Microsoft Visual Basic 6.0. Sistem ini telah diuj i dalam persekitaran akademik (Universiti Utara Malaysia). Bagi meningkatkan keupayaan fungsi dan ciri-ciri sistem ini, pelbagai usaha seperti menyempumakan sistem ini kepada satu sistem yang lengkap dan memperkembangkan sistem ini kepada satu sistem sokongan pemutusan secara kumpulan (GDSS) dapat dilakukan pada masa depan.

ABSTRACT (ENGLISH)

This project is aimed at developing a low cost and end user oriented financial planning decision support system (DSS) for small medium enterprise (SME) in Malaysia. Low cost and user friendly DSS is essential to assist the SME decision makers in their future financial planning as the available financial planning software are too expensive and not compatible for the SME in Malaysia. An integration of Microsoft Excel 2000 and Microsoft Visual Basic 6.0 had been used to develop this system. The financial models had been developed using Microsoft Excel 2000 (spreadsheet based) and the user interfaces had been developed using Microsoft Visual Basic 6.0. Upon the system prototyping process, the system had been tested in an academic environment (Universiti Utara Malaysia). Further works such as extending the system to a full functional system and a group decision support system (GDSS) could be done to improve the functionality and feature of the system.

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my supervisor, Associate Professor Nazib Nordin for his constructive suggestions, guidance, and consistent support during the developing project process. Secondly, I would like to thank Graduate School of UUM for giving me the chance to pursue my master degree at Universiti Utara Malaysia.

Lastly, I am greatly indebted to my family for their sacrifice during my study.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

AHP	Analytical Hierarchy Process
DBMS	Database Management System
DSS	Decision Support System
GADS	Geodata Analysis and Display System
GDSS	Group Decision Support System
GUI	Graphical User Interface
IDSSBALM	Interactive Decision Support System For Bank Asset Liability Management
KS	Knowledge System
LS	Language System
MAMD	Multiattribute Modeling Approach
MBMS	Model Base Management System
MCDM	Multiple Criteria Decision Making
MINORA	Multicriteria Interactive Ordinal Regression Analysis
MSI	Medium-Scale Industry
PPS	Problem Processing System
PRM	Three Parameter Risk Measure
PS	Presentation System
SME	Small Medium Enterprise
SMI	Small-Scale Industry
TSI	Tiny-Scale Industry
UIMS	User Interface Management System

Chapter 1

INTRODUCTION

The use of computer-based systems in supporting the management decision-making process has increased dramatically over the past decades. But, the development of the low cost and end user oriented financial planning DSS is still an issue that relatively receives little attention. Consequently, this project is denoted to develop a low cost as well as end user oriented financial planning decision support system (DSS).

This chapter gives an overview of Small Medium Enterprise (SME) in Malaysia and Decision Support System (DSS) in financial area.

The problem statement, objectives, project scope, hardware and software requirements of this project are discussed.

1.1 Decision Support System (DSS)

According to Turban and Aronson (1998), DSS is a system intended to support managerial decision makers in semi-structured decision situations. DSS is meant to be an adjunct to decision makers, to extend their capabilities but not

to replace their judgment. It can also be considered as a model-based set of procedures for processing data and judgment to assist upper management in his or her decision-making.

Since the purpose of the DSS is to aid the top management in decision-making, Turban and Aronson (1998) has concluded that the DSS must be a computer-based system consisting of three interacting components. The interacting components are a language system, a knowledge system, and a problem-processing system.

A language system is a mechanism that provides communication between the users and other components of the DSS. A knowledge system is a repository of problem domain knowledge embodied in the DSS, either as data or procedures. A problem process system is the link between the language system and knowledge system, containing one or more of the general problem-manipulation capabilities required for decision making.

For the financial planning purpose, DSS model is most properly model based and algebraically oriented. This means, the algebraically formulas are written in the manner in which one would write mathematical equation. As such, spreadsheet-oriented DSSs are in widespread use today.

1.2 Small Medium Enterprise (SME) In Malaysia

In Malaysia, there are several characteristics to recognize the firms of SME. Generally, SMEs in Malaysia can be categorized into three groups. They are the tiny-scale industry (TSI), small-scale industry (SMI), and medium-scale industry (MSI). TSI is defined as the firms that employ 4 employees or less. While, the SMI can be differentiate as the firms that hire employees between 5 to 49 persons. Lastly, the MSI may have between 50 to 199 employees (Lim Chee Peng 1986: 17-33).

The capital amount of the firms is a measurement to differentiate large-scale industry from the SME. The SME invests a relatively small proportion of the total capital compare to the large-scale firm. Since the SME has low level of capitalization, the firm is usually organized as a family or sole proprietorship business, a reflection of the small-scale operation of the enterprise.

1.3 Problem Statement

The DSS is important to assist the SME decision maker to make an accurate financial planning decision based on the relevant and reliable data and information. Since the financial resources for SME are small, spreadsheet based DSS is the cheapest system to gain precise and accurate decision compare to the complicated and costly software (e.g. Visual IFPS / Plus and

ENCORE Plus!). For this reason, the spreadsheet-based DSS is suggested to solve the financial problem of the SMEs in Malaysia.

Although the spreadsheet-based DSS is cheaper than other financial planning software, the decision maker may face the problem of using it. According to Turban and Aronson (1998), decision maker may encounter two major difficulties with spreadsheet: formulas are hard to decipher, and time sequences are difficult to handle. The user interface of a spreadsheet based DSS has to be improved to make it more applicable in the SME environments.

1.4 Objectives

The main objective of this project is to develop a financial planning decision support system (DSS) suitable for Small Medium Enterprise (SME) in Malaysia. This project aims at achieving the following features:

- To develop a spreadsheet- based DSS for SME financial planning.
- To develop a model based DSS.
- To develop an interactive DSS so that the decision makers can use the program with ease. This interactive DSS will include the graphical user interface, help facilities, versatility ability, uniformity, consistence, and informativeness.

- To develop a DSS to aid the decision makers to optimize and forecast the financial planning decision of SME.

1.5 Project Scope

This project will focus on the design of a dialog and model components of DSS. The models to be developed are based on the following financial statements:

- Balance Sheet
- Capital Budgeting
- Loan Payment Analysis
- Operating Budget
- Cash Flow (Under Constructions)

With the user-friendly interface, this project will provide useful financial analysis information for the SME decision makers.

1.6 Software And Hardware Requirements

Hardware requirements for this project are as below:

- A Pentium-based PC-compatible computer system
- 32 MB of RAM (64 MB recommended)
- 400 MB of disk space

- An SVGA-compatible display (16bit or more colors recommended)

Software requirements for this project are as below:

- Microsoft Windows® 95 or Windows® 98
- Microsoft Excel 2000
- Microsoft Visual Basic 6.0

1.7 Summary

Due to limited financial resources faced by the SMEs, a spreadsheet based financial planning DSS is chosen as a solution tool for the SMEs. However, spreadsheet based DSS are not user friendly. It is the purpose of this project to overcome the problem of user interface in spreadsheet based DSS. The DSS to be developed will more applicable in the SME environments.

Chapter 2

STATE OF TECHNOLOGY

This chapter reviews the generic framework of a DSS, classifications of DSS, components of DSS, and applications of DSS in financial areas.

The applications of DSS will be explained in terms of the architecture or framework, information requirements, model designed and technology applied in the financing institutions, cash managements, and bank asset liability management. Lastly, the comparisons will be made among the existing DSS applications.

2.1 Technology of Decision Support System

Decision Support Systems are computer-based systems that help decision makers to confront ill structured problems through direct interaction with data and analysis models.

As illustrated in figure 2.1, the user will select a desired element of the LS (Language System) that consists of all messages the DSS can accept. As the software engine of DSS, the PPS (Problem Processing System) will process the LS element. During the processing, the PPS will manipulate the KS

(Knowledge System) contents or change the knowledge that held in the KS. In either event, the PPS will issue a response to the user. Lastly, the PS (Presentation System) will present the response to the user.

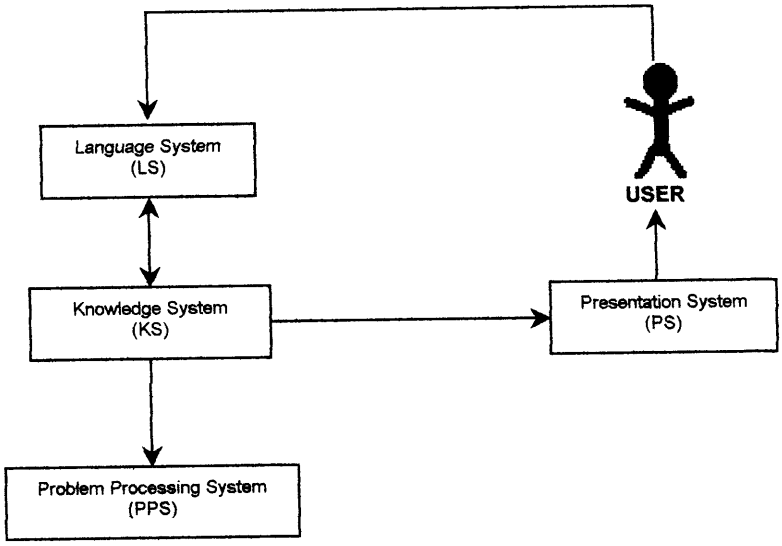


Figure 2.1: Generic Framework of A DSS

2.1.1 Classifications of DSS

DSS is classified into three levels of technology, namely, specific DSS, DSS integrated tools, and DSS primary tools. Figure 2.2 shows the levels of the technology.

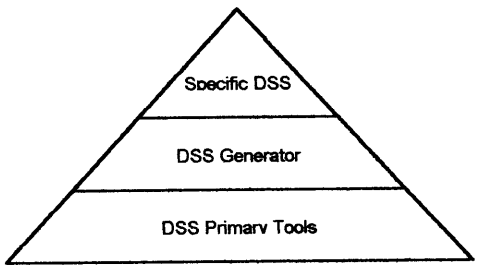


Figure 2.2: Levels of The DSS Technology

Each of the technology level of DSS is described according to figure 2.2:

- *DSS Primary Tools*

  The DSS primary tools are the lowest level of DSS technology. These tools facilitate the development of a DSS generator and a specific DSS. Examples of DSS primary tools are programming languages, graphics, editors, query systems, and random number generators.

- *DSS Generator*

  The DSS generator is also known as DSS engine. It is an integrated development software package that provides a set of capabilities to build a specific DSS quickly, inexpensively, and easily. Example of a DSS generator is Microsoft Excel 2000.

- *Specific DSS (DSS Applications)*

  A specific DSS or DSS application is a final product that actually accomplishes the work.

2.1.2 Relationships Among The Technology Levels

The DSS primary tools are used to construct generators, which in turn are used to construct specific DSS. It can also be directly used to construct the specific DSS.

However, the DSS generators are used to construct the specific DSS and enable the specific DSS to adapt quickly to the changes (Turban and Aronson, 1998: 280).

2.1.3 Comparisons Between DSS Generators And DSS Primary Tools

The comparison between the DSS generators and DSS primary tools can be made in terms of cost and time aspects.

- *Cost*

  The DSS generators are cost saving compare to DSS primary tools. This is because the DSS generators enable the specific DSSs to adapt quickly to changes.

- *Time*

  The DSS generators are time saving compare to DSS primary tools, as DSS primary tools must be developed when it is needed (Turban and Aronson, 1998: 280). However, the DSS generators are available in the market in an easy-to-use package, such as Microsoft Excel 2000 and Lotus 1-2-3.

2.2 Components of DSS

According to Turban and Aronson (1998), DSS is composed of data management subsystem, model management subsystem, knowledge management subsystem, and user interface (dialog) management subsystem.

Figure 2.3 illustrates the relationships among the components of A DSS. A user will make a request through the user interface management subsystem. Later, the request will be sent to data management subsystem, model management subsystem, or knowledge management subsystem. After the request being processed in one of the management subsystems, response will be sent to the user through the user interface management subsystem.

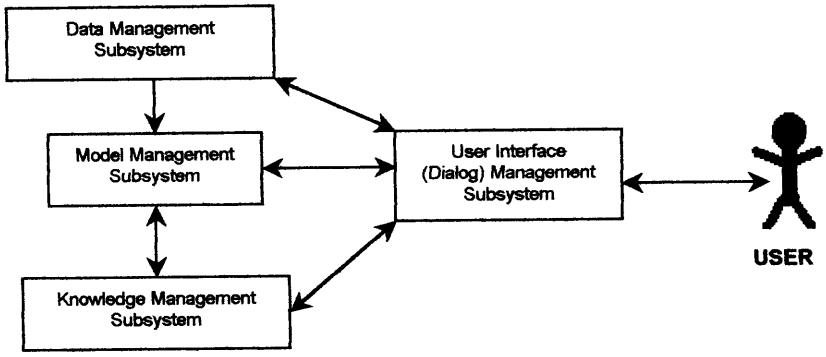


Figure 2.3: The Relationship Among The DSS Components

2.2.1 Data Management Subsystem

The data management subsystem includes the database that contains relevant data for the situation. It is managed by the database management system (DBMS).

Figure 2.4 illustrates the relationships among the components of the database management system, which goes through the following steps.

Step 1

The database management subsystem will start when the user make a request through query facility. The query facility allows the users to access, manipulate, query the data use in the DSS.

Step 2

The request is sent to the database management system (DBMS). The DBMS allows user to create, update, or access data. Later, the data requested is sent to the DSS database.

DSS database is a collection of interrelated data that organized to meet the needs and structure of an organization.

Step 3

External sources from external or internal database are sent to the DSS database to support the processing activities.

The data directory supports all the activities such as adding new entries, deleting entries, retrieving information from the DSS database. It acts as a catalog or reference of all data in the DSS database.

Step 4

After the processing activities, the DSS database will send the responses to the user through the DBMS and query facility.

As a conclusion, the query facility is the only component that enables the users to communicate with or command the DSS in the data management subsystem.

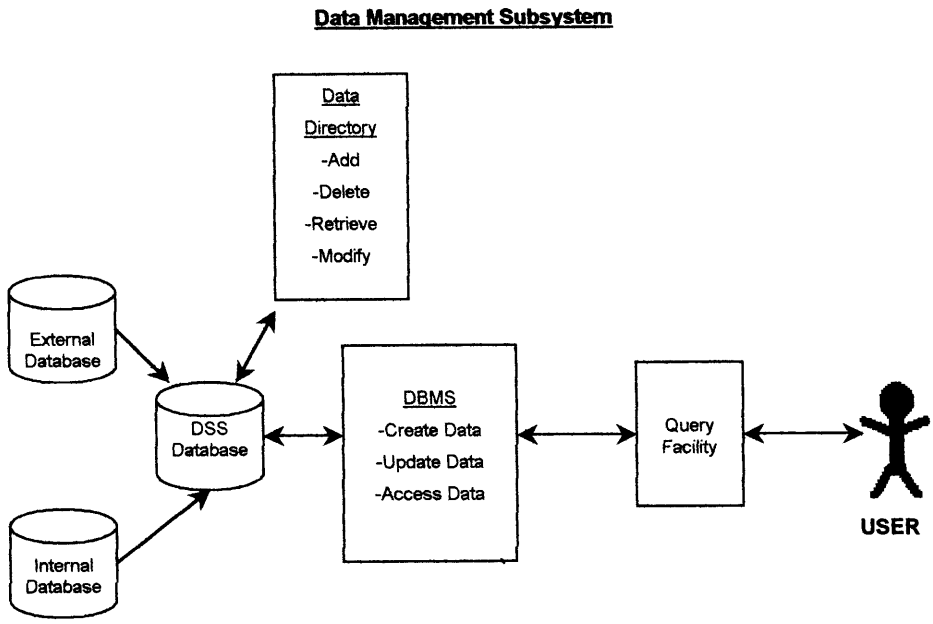


Figure 2.4: The Relationships Among The Components of Data Management Subsystem

2.2.2 Model Management Subsystem

The model management subsystem provides the system analytical capabilities through financial, statistical, management science, and quantitative models.

Figure 2.5 shows the architecture of the model management subsystem. The processes for the model management subsystem are described as below.

Step 1

The model base management subsystem will start when the user makes a request through the modeling language. Later, the request will be sent to the MBMS (model base management subsystem).

Step 2

To start a processing activity, the MBMS will send the request to the model base through a database. According to Turban and Aronson (1998), the MBMS is capable of interrelating models with the appropriate linkages through a database.

Step 3

Model base provides the ability to invoke, run, change, combine, and inspect the models components. The models components are needed to analyze and process the request in a DSS. Examples of the models components are routine and special statistical, financial, forecasting, management science, and other quantitative models.

Here, a model directory is used as a catalog for the models and other software in the model base.

Step 4

After the processing activities, the DSS will send the responses to the user through the MBMS and modeling language.

As a conclusion, the modeling language is the only component in model base management subsystem, which allows the user to communicate with or command the DSS.

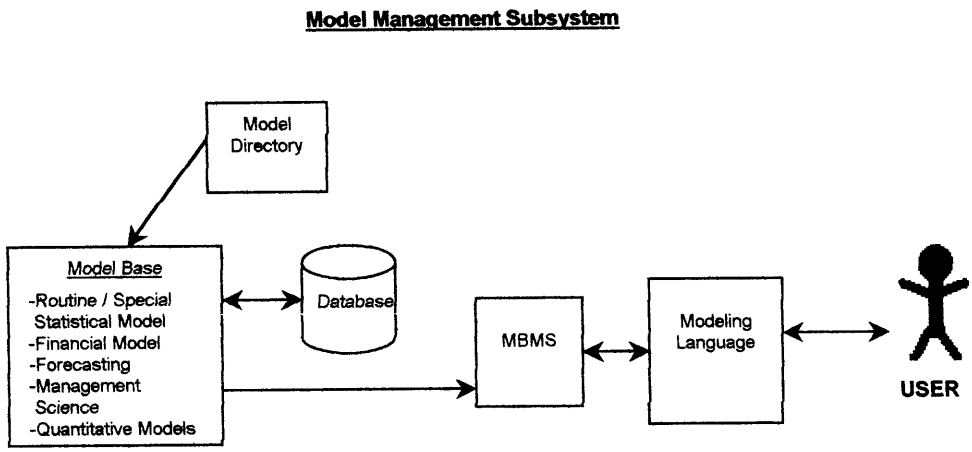


Figure 2.5: The Relationships Among The Components of Model Management Subsystem

2.2.2.1 Review of Model Base Subsystem

A model is an abstract representation of the essential elements of a real world phenomenon. There are many types of models, including verbal, schematic, analog, and mathematical. The models represent different degrees of concreteness or abstractness (Todd and Bennasat 1996: 241). At each stage

of the methodology, the model will guide through problem solving processes.

The model domain is spanned from planning, financial, forecasting, software engineering, algebra word problems, engineering design, and the likes (Sen and Vinze 1996: 239). The financial planning model is based on the mathematical structures. For instance, financial planning models may entail a collection of accounting formulas (mathematical equations) for producing pro forma statements.

The model based financial planning focuses on the design of a normative model that allows the decision maker to structure various financial planning requirements, variables, and constraints. The model can be a black box, where the decision makers do not understand the algorithm and solution procedures. However, this model is able to generate and compare the alternative scenarios in fairly simple and transparent terms (Hokey Min 1989: 208)

A study done by Hokey Min (1989) regarding model based decision support system for locating banks has used the GADS (Geodata Analysis and Display System) and MCDM (Multiple Criteria Decision Making) models to manipulate the model base management subsystem.

- GADS is used to identify the potential market boundaries that determine a limited number of potential bank sites and measure the future market share.

- MCDM model is used to capture the realistic aspects of the bank location decisions as well as its usability (Hokey Min 1989: 210). In MCDM model, every parameter and constraint is defined to formulate and evaluate the models.

The study has found that the traditional normative approach is not able to analyze the excessive location-related data that associated with the external factors. It is also unable to evaluate the behavioral criteria for the branch bank sites. To overcome this difficulty, the study has recommended model based DSS. For this DSS, the model formed is embedded within a locational decision support system for branch banks. It is an interactive fuzzy goal-programming model.

2.2.3 Knowledge Management Subsystem

According to Turban and Aronson (1998), the knowledge management subsystem can support any of the other subsystems or act as an independent component.

Knowledge management is used when it needs to integrate with typical DSS, supports uncertain decision that is not supported by the typical DSS, and perform as a library for the DSS models. When the knowledge management is applied in a DSS, the DSS is said to have the intelligence ability. It can be recognized as expert system or knowledge base DSS.

Typical DSS refers to DSS that consists of database management subsystem, model base management subsystem, and dialog management subsystem.

2.2.4 User Interface (Dialog) Subsystem

The user interface or dialog subsystem allows user to communicate with and command the DSS (Turban and Aronson, 1998: 79). This subsystem is managed by the user interface management system (UIMS). UIMS enables the user to interact with the model management and data management subsystem. In advanced systems, the user interface components include a natural language processor and GUI (graphical user interface).

2.2.4.1 Review of User Interface Subsystem

A review of literature reveals that prior models of the DSS user interface are oriented largely toward programmers and system designers (Sankar et. al 1995: 93). Hence, many DSS user interfaces are inflexible and difficult to use.

Designers were more concerned with the programming in the internal architecture of the DSS. Most of the research in DDS focuses on data, procedures, rule sets, text forms, and spreadsheets associated with the problem or decision area. As a result, the user interface is frequently ignored.

A quality or friendly interface will assist the user to learn and use the system with ease. Poor quality user interface is a major reason why managers have not used computers and quantitative analysis (Sankar et. al 1995: 94).

According to Sankar et. al (1995), the features to have an adaptable interface or a good quality system are: inputs and outputs of a user's queries can be displayed on screen or presented in graphical or textual form.

Jones (1995) has concluded that graphs are widely used in DSS to represent the complicated problems. Graph-grammars provide a theoretically grounded, practical tool for building user interfaces and some degree of functionality for problem domains. A problem holder needs a DSS to support a particular class of decisions. Although the problem holder may not have a precise description or model in mind, the problem holder believes that some form of picture, in particular a graph, can represent the problem at hand as well. The information presented in graphs may give the problem holder a clearer idea of things.

Capturing user's requirement is the most important indicator in developing an interface. The developers must ensure that the users must be familiar with the system and the problems or decisions being supported by the system. This is because the users are the people who will utilize the information produced by the system.

2.3 DSS Application In Financial Planning

Financial planning DSS is been widely used in the commercial sectors. The use of the DSS is mostly specific to a particular area such as cash management, credibility evaluation, and the likes.

Below is a review on the DSS applications in the financing institutions, cash managements, and bank asset liability management. The review will be made along with the architecture or framework of a DSS application, technology and model design, and information required to run the DSS application.

2.3.1 Integrated DSS For Financing Firms

Siskos et. al (1994) presented an integrated DSS for analysis and financing industrial in Greece. Specifically, this integrated DSS is developed for a Greek Industrial Development Bank (ETEVA) to evaluate the viable firms that intend to apply for financial aids.

Mainly, the integrated DSS is aimed to assist the financial organizations (credit institutions, banks) in making decisions to finance the viable firms.

It also aim at (Siskos et. al 1994: 152):

- Forecasting and preventing difficulties of the bank, consequently, eliminating high risk in financing operations.
- Upgrading financial art.
- Supporting the managerial personnel of firms.

To assist the top personnel in selecting a stable firm, the DSS aims at:

- Forecasting and preventing the difficulties of the potential financing firms.
- Eliminating the high risk in financing operational such as participation in capital (Siskos et. al 1994: 153).
- Giving supports to the managerial personnel of the firms.

2.3.1.1 Framework of Integrated DSS For Financing Firms

The integrated DSS for financing firms consists of database subsystem, model base subsystem, and dialog subsystem.

Figure 2.6 illustrates the framework of the integrated DSS for the financing firms in Greek.

The dialog system enables the user to communicate with or command the DSS. However, the model based system provides financial analysis to user. The financial analysis includes common size statements, financial ratios, and graphics. Two mathematical models have been applied in financial analysis. The mathematical models are multivariate statistical methods and corporate risk models.

The database system supplies the financial data to the model base system. The financial data is taken from the pro forma statements, namely, balance sheet and income statement. To ensure the accuracy of the analysis, qualitative information is included. The qualitative information consists of information security, market trend, market niche or position, and quality of management.

After the processing activities in model based system, the DSS will send responses to the user through the dialog system.

As a conclusion, the dialog system is the only components that communicate with user. And, the model base system is the software engine of the DSS. Finally, the database system is the record store for the DSS.

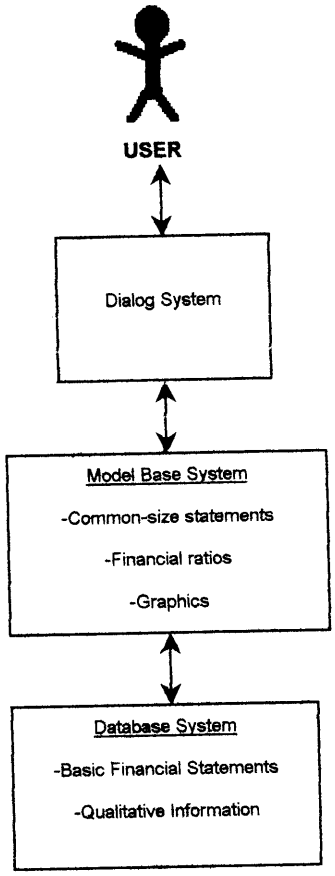


Figure 2.6: Framework of Integrated DSS For Financing Firms

2.3.1.2 Information Requirements of Integrated DSS

The system evaluates the financial performance of firms in terms of financial ratio of profitability, managerial performance, and solvency during a 5-year period (Siskos et.al 1994: 151).

The study reveals that the model base system includes (Siskos et. al 1994: 154):

- Financial analysis
- Corporate risk models

Each of the model and information requirements in the integrated DSS is presented in table 2.1.

Model	***Information Requirement***
Financial analysis	-Financial ratios for a 5-year period
Corporate Risk Models • Principal components analysis • Discriminant analysis	-Financial ratios for a 5-year period

Table 2.1: Models And Information Requirements of Integrated DSS (Adapted from Siskos et. al 1994)

2.3.1.3 Technology And Model Design of Integrated DSS

The integrated DSS for financing firms in Greek runs on IBM compatible machines equipped with a graphic card. It needs a mathcoprocessor to speed up the portions of the systems (Siskos et. al 1994: 157). Microsoft's

Professional Development System 7.0 (Quikbasic environment with support toolboxes for graphic and user interface) has been used to develop this DSS.

The financial models and corporate risk models are formed using the multivariate statistical method and MCDM (mutlicriteria decision making methods). For the MCDM, MINORA (acronym of Multicriteria Interactive Ordinal Regression Analysis) system is used in the DSS to form the corporate risk models. In MINORA system, the ordinal regression or a linear programming formulation has been used.

As a conclusion, a DSS primary tool (a linear programming formulation) has been used in this integrated DSS.

2.3.2 DSS For Integrated Cash Management

According to Srinivasan and Yoong H. Kim (1986), the DSS for integrated cash management is aimed to recognize the interrelationships among the sub-problems as well as between the sub-problem and other financial decisions.

This DSS has helped the cash managers and corporate treasurers to overcome the deficiencies in forecasting or predicting the sub-problems such as fluctuation of interest rate that occurred in the cash management.

The decision support for integrated cash management is a framework that only recognizes the typical normative approach in cash management. Here, the normative approach refers to the modeling methodology that forms the basic support model in the DSS (Srinivasan and Yong H. Kim 1986: 349).

The integrated cash management will be described in terms of decision process, model base design, technology used, and information requirements.

2.3.2.1 Decision Process of Integrated Cash Management DSS

The decision process of integrated cash management DSS begins from the external database, where the data of information requirements are stored.

The decision maker has five alternatives, namely, cash balance management, cash gathering, cash mobilization and concentration, banking system design, and cash disbursement.

After selecting the alternative, the decision maker either make operational decisions or infrastructural decisions. For operational decisions, decision maker can send the decisions to operational DSS or analysis operations. However, for infrastructural decisions, the decision maker can send the decisions to infrastructural DSS or analysis operations.

The analysis process will end when the decision maker gets a feasible solution. If not, the decision maker can continue the process loop until a feasible decision is provided.

Figure 2.7 shows the decision process of the integrated cash management DSS.

2.3.2.2 Technology And Model Design For Integrated Cash Management

For the integrated cash management, the four major decision types use the mathematical programming tools to form the models in the DSS. For instance, the cash balance management subsystem has used the AHP (Analytical hierarchy process), member of the MAMD (Multiattribute modeling approach) to form cash management models.

As a conclusion, the technology tools in this DSS consist of DSS primary tools (mathematical programming formulations).

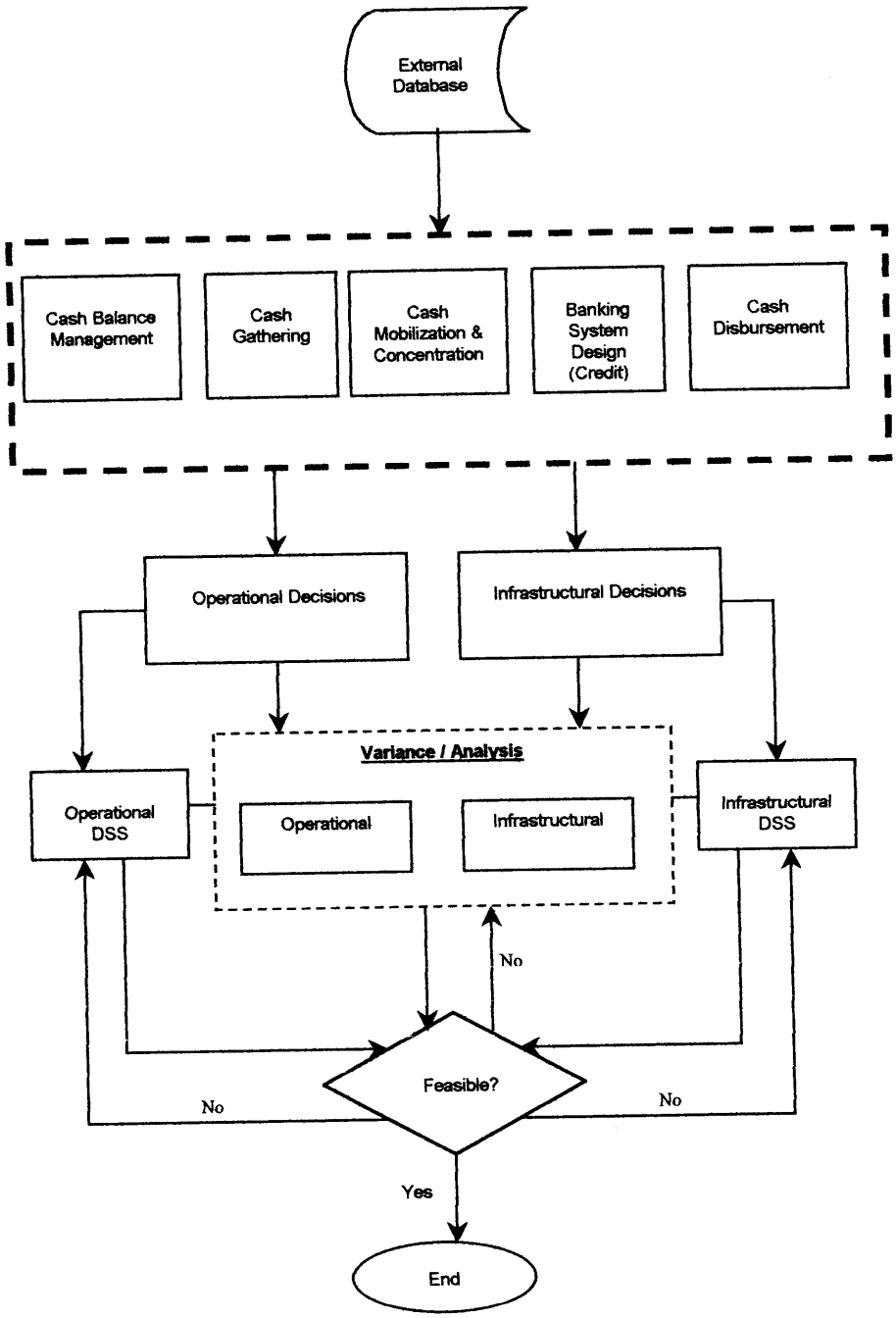


Figure 2.7: Decision Process of Integrated Cash Management DSS

2.3.2.3 Information Requirements For Integrated Cash Management

The study reveals that the integrated cash management can be decomposed into five major decision types (Srinivasan and Yong H. Kim 1986: 350):

- Cash balance management
- Cash gathering
- Cash mobilization and concentration
- Cash disbursement
- Banking system design for credit services

With these major decision types, a number of subproblems such as cash forecasting, lockbox location, disbursement system and the like can be identified.

Table 2.2 summarizes the major decision types, several subproblems and information requirements that are needed in the integrated cash management.

Major Decision Types	*Subproblems*	*Information Requirements*
Cash Balance Management	-Cash position management -Cash forecasting -Investment of surplus cash -Borrowing to meet cash deficit	-Information on balances -Forecasts of receipts and disbursement -Fund transfers -Details of company policies, strategies, constraints -Financial market information
Cash Gathering	-In-house processing -Lockbox location	-Information concerning mail and availability times relating to a group and collection of a center for the existing system -Total amount of incoming funds form each group -Fixed and variable costs associated with the firm's present and proposed systems for processing a group's checks through a related collection center -Firm's cost of capital -Information regarding actual mail and clearing times, actual volume and dollar amount of checks processed through various collection centers
Cash mobilization And Concentration	-Design of concentration systems -Choice of fund transfer mechanisms	-Information on deposit and disbursement banks -Potential concentration banks and alternative designs -Cost and benefit information on alternative designs

	-Cash transfer scheduling	-Information concerning mail and availability times from deposit banks to concentration banks and from concentration banks to disbursement banks for each of the alternatives -Costs and availability times for each of the transfer mechanisms -Non-financial benefits form banks and transfer mechanisms -Collection and disbursement data
Cash Disbursement	-Disbursement system -Disbursement techniques	-Comparative cost and benefit information on the alternative banks -Forecast information on expected cash flows

Table 2.2: Summary of The Major Decision Types, Several SubProblems And Information Requirements (Adapted from Srinivasan and Yong H. Kim 1986)

2.3.3 DSS For Bank Asset Liability Management

The IDSSBALM (acronym of Interactive Decision Support System For Bank Asset Liability Management) is aimed at maximizing the bank's gain or returns, bank's balance or business volume, minimizing the bank's credit losses, and lastly, minimizing the interest rate risk.

According to Langen (1989), there are two versions of IDSSBALM. For the current version, users need to use programming command in order to communicate with the DSS. Consequently, the future version is built to improve the inflexibility of the current version of IDSSBALM.

There are two types of users of the future version of IDSSBALM. They are:

- The sophisticated user (financial analyst) who knows the influence of the different parameters and wants to change the parameters according to his or her needs.

- The merely practical bank oriented user that do not have programming skill.

The future version of IDSSBALM is also aim at creating and building a model and decision support system that is able to comprise and to consider the most important quantifiable bank objectives. It also aims to provide the users with one or several solutions as a basis for further decision making (Langen 1989: 389).

2.3.3.1 Conceptual Architecture of IDSSBALM

The conceptual architecture of IDSSBALM consists of database, system users, model builders and decision maker input raw data. There three steps to describe the conceptual architecture of IDSSBALM.

Figure 2.8 shows a conceptual architecture of the IDSSBALM. The processes of the conceptual architecture are explained as below.

Step 1

The decision maker either a financial analyst or a bank manager could specify his or her models and problem structure by choosing the submodels provided. The submodels are built according to objectives, risks, and constraints of the bank.

Here, the financial analyst refers to interpreters, intermediary who are trained in mathematical programming (Langen 1989: 398).

Step 2

In IDSSBALM, mathematical programming formulation is used to run the submodels. IDSSBALM also contains some interactive command types, namely, PRINT, RESULT, START, OTHER and the likes (Langen 1989: 398).

The user's request will be processed in IDSSBALM using the submodels provided.

Step 3

Before sending the responses to the user, IDSSBALM may send the responses to a database as a record keeping to the bank's departments. IDSSBALM may also ask for raw data processing from the database. Lastly, the IDSSBALM will send the responses to the user.

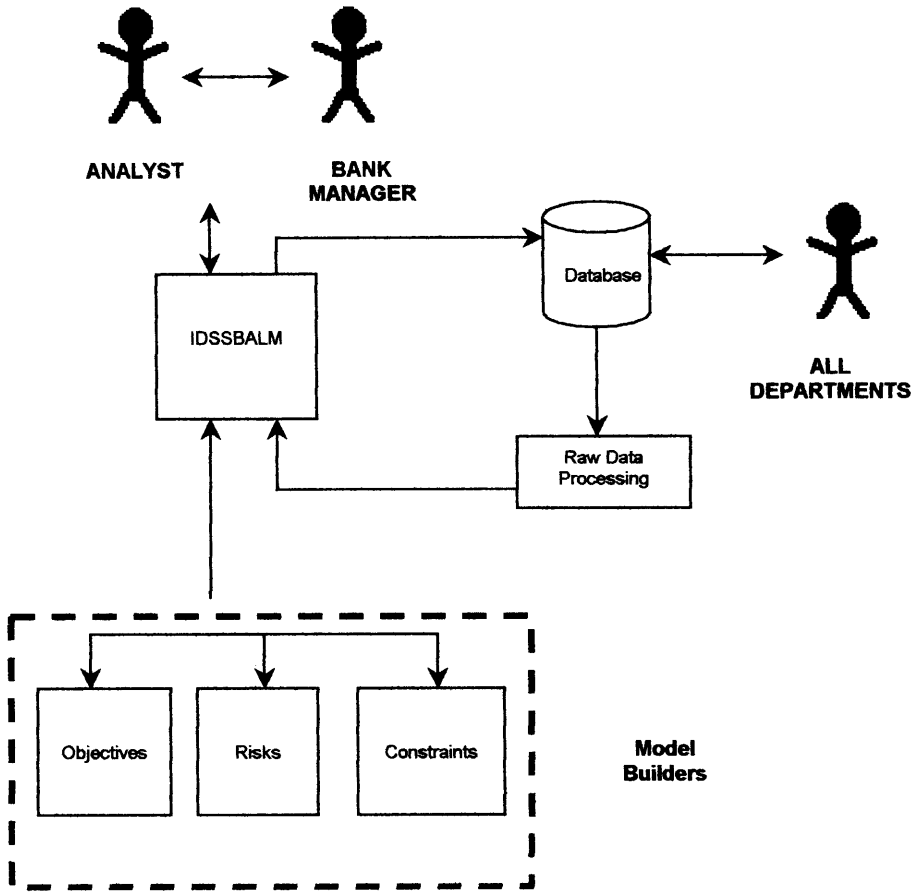


Figure 2.8: A Conceptual Architecture of IDSSBALM (Adapted from Langen 1989)

2.3.3.2 Information Requirements of IDSSBALM

Information requirements of IDSSBALM depend on the models used and sub-elements of the models. Table 2.4 shows the models, subelements, and information requirements of IDSSBALM.

Models Builder	***Subelements***	***Information Requirements***
Objectives	-Maximize the expected returns on interest business -Maximize the balance volume -Minimize the expected credit risks	-Bank's internal policy -Foreign currency exchange rates -Overall current risk over the planning horizon -Balance sheet ratio analysis -Information about the speculations made
Risks (unipolar or bipolar)	-Expected utility -Stochastic dominance -Probability dominance -3-PRM (Three parameter risk measure) -Bipolar risk measures	*Unipolar* -Utility functions -All possible asset or liability portfolio alternatives -The expected value function of the corresponding uncertain function -Other target risk *Bipolar* -Negative deviations from EV (the actual risk to be minimized) -Positive deviations from EV (possible changes)

Constraints	-Germany banking law principle I, II, and III -Cash position -Balance equation -Managerial balance structure constraint	-Market constraints based on forecasts and management politics

Table 2.3: Models, Subelements, And Information Requirements of IDSSBALM (Adapted from Langen 1989)

2.3.3.3 Technology And Model Design For IDSSBALM

IDSSBALM is implemented on a VAX 8900 in FONTRAN (Langen 1989: 398). The models of IDSSBALM cannot be used in the PC-based environment. This is because the models used are too large for the PC version.

As a conclusion, IDSSBALM is a specific DSS as it is tailored to a special problem environment (bank asset / liability management).

2.4 Comments On The Existing Financial DSS Applications

From the review above, the existing DSS applications are slanting to the use of the large-scale industry such as financial institutions, banking industry, and the likes. Thus, the costs of the existing financial DSS applications are

more expensive as these DSS applications aim at large scale industry. This circumstance causes the small and medium industries unable to afford for the existing financial DSS. Hence, the DSS generator is a feasible solution in overcoming this obstacle.

Microsoft Excel 2000 is a powerful DSS generator to remodel the expensive financial DSS applications for the SME. But, the user interface of the spreadsheet in Microsoft Excel 2000 is not appropriate, as the SME decision makers need to build the financial models when working with the DSS generator (spreadsheet). To overcome this difficulty, the user interface for the DSS generator needs to be improved.

Table 2.4 shows the comparisons among the existing DSS applications. The comparisons are made in terms of technology, target user, and information requirement of the DSS applications. The comparisons are also made concerning the time length and cost spent to build the DSS application.

2.5 Summary

Generally, the framework of a DSS consists of a language system (LS), problem processing system (PPS), knowledge system (KS), and presentation system (PS). As well, a DSS also consists of four components, namely, database management subsystem, model base management subsystem,

knowledge management subsystem, and user interface management subsystem.

Ordinarily, a DSS can be developed with the technology, namely, DSS primary tools, DSS generators, or specific DSS. However, the review has shown that the DSS generators especially the spreadsheet based managements have the advantages in terms of cost and time saving compare to the DSS primary tools. Also, the DSS applications used in the financing institutions, cash managements, and bank asset liability managements have implied that the DSS is useful in the financial areas.

Anyhow, the spreadsheet based DSS does not have a user-friendly interface. This is because the users need to form the mathematical formulations in the grid cells. For this reason, the user interface of financial planning DSS needs to be improved.

DSS Application	*Technology*	*User Segmentation*	*Information Requirements*	*Cost*	*Time Spent*
Integrated DSS For Financing Firms	DSS primary tool (a linear programming formulation)	Financing firms or banking industry	Low	High	High
Integrated Cash Manage-ment	DSS primary tools (mathematical programming formulations).	Corporate firms	High	High	High
IDSSBALM	Specific DSS	Banking industry or finance institutions	High	High	High

Table 2.4: Comparisons Among The Existing DSS Applications

Chapter 3

PROJECT METHDOLOGY

This chapter will give an in-dept discussion on the project methodology. This methodology will adapt the prototyping process with the DSS based methodology comprising of planning, system requirements, design, prototyping, and testing and evaluation phases as shown in figure 3.1.

3.1 Planning

Planning deals with need assessment, problem diagnosis and objectives. A crucial step in this planning effort is to determine the key decision to be supported by the system (Turban and Aronson 1998: 269).

3.2 System Requirement

This phase involves the identification of a relevant approach for addressing user needs and review of related literature. The use case is used to identify the user needs and system functionality.

Use case diagram is used to show the functionality that the system will provide and which users will communicate with the system in some way when it provides that functionality (Bennett et. al 1999: 111).

3.3 System Analysis

This phase includes the determination of the best construction approach and specific resources required to implement the system. Basically, the feasibility study will be done followed by the conceptual design.

3.4 System Design

The detailed specifications of the system components, structures, and features are determined. The design can be divided into model based subsystem and dialog subsystem.

Mainly, the model-based subsystem in this system is generated from the financial models.

The dialog subsystem allows the user to communicate with the DSS. It enables user to command the DSS. For this project, Microsoft Visual Basic 6.0 has been used to develop the system user interface.

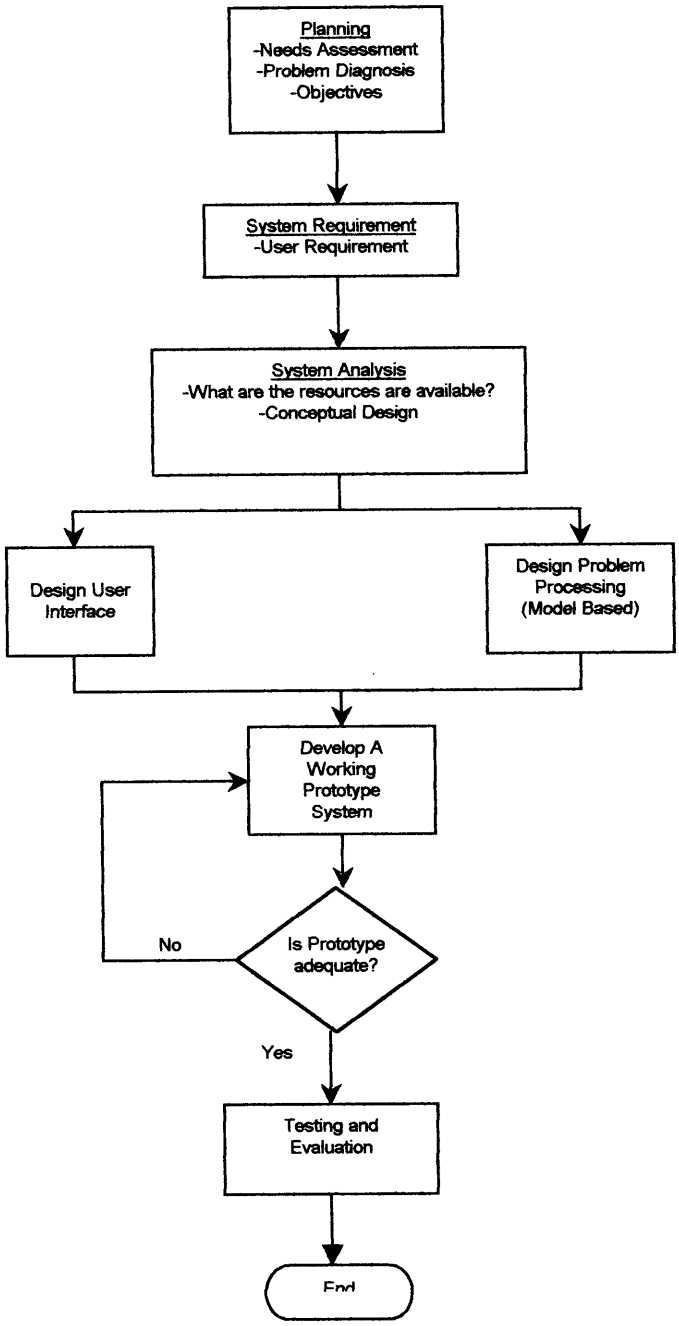


Figure 3.1: Project Methodology Phases

3.5 Prototyping

The prototyping approach aims at building a DSS in a series of short steps with immediate feedback from users to ensure that the development is proceeding correctly (Turban and Aronson 1998: 272).

Because of the semi-structured nature of problems addressed by DSSs, manager's perceived needs for information may be unclear (Turban and Aronson 1998: 272). Therefore, an adapted prototyping process is been used to develop this project.

3.5.1 Prototyping Process

Table 3.1 gives a brief description of each prototyping process.

Prototyping Process	Description
Identify important sub-problems to be built	This process will adopt the planning phase.
Develop a small and usable system	This process will adopt the system requirements, system analysis, and design phase.
Test and evaluate the system process	This process is to verify that the system does meet the stated functions and requirements.

Documentation	The user and system manuals will be created in this process.

Table 3.1: Prototyping Process

3.5.2 Time Box of Prototyping Process

Table 3.1 shows the prototyping process and the time box iterations.

Prototyping Process	Time Box Iterations
Identify important sub-problems to be built	
Develop a small and usable system	
Test and Evaluate the system process	
Documentation	

Table 3.2: Time Box of Prototyping Process

3.6 Testing And Evaluation

This system is tested in an academic environment. It is tested through the functionality and feature of the system. Figure 3.2 shows the testing and evaluation process of the system.

The system functionality is tested in term of the capabilities of the system. However, the system feature is tested in term of the interface characteristics.

The user characteristics are tested through:

- **The Length Of Time**

The interface of the system is tested through the length of time taken by the user to perform the system. To ensure the results across the users comparable, all users are instructed to do the same operations (DeSantics et. al 1994: 322).

To evaluate this system, the user is asked to enter only the numeric data and select the subsystem in the option menu. The length of time will be taken when the user begin the operations mentioned until the user stops the operations.

- **Error Rate**

To evaluate this system, error rate is calculated by taking the total number of errors committed by the user in performing the operational task and dividing to the length of time mentioned above.

The average across all users is the overall error rate. Variance in time and errors committed provide an indication of the range responses that people may have in learning the system (DeSantics et. al 1994: 322).

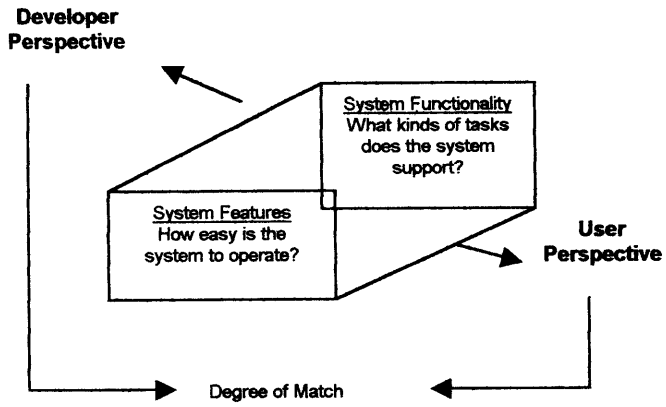


Figure 3.2: Testing And Evaluation Process of The System

3.7 Summary

The project methodology is an adapted version of prototyping process and DSS based methodology. The phases in the methodology involved are planning, system requirements, system analysis, system design, prototyping, testing and evaluation.

The prototyping begins with a sub-problem where the steps in the planning phase have been adopted in the process. A small system will be developed adopting the system requirements phase, system analysis phase, and design phase. The system will be tested and evaluated to verify that the new version system does meet the stated functional specifications and requirements.

Chapter 4

SYSTEM REQUIREMENT ANALYSIS

This chapter will recap the objectives of the system. Following this, the use case model is used to identify the system actors and the task of each actor. Later, the system requirements in terms of functionality and feature will be discussed.

4.1 Objectives

The main objective of this project is to develop a financial planning decision support system (DSS) for Small Medium Enterprise (SME) in Malaysia. This project aims at achieving the following features:

- To develop a spreadsheet- based DSS for SME financial planning.
- To develop a model based DSS.
- To develop an interactive DSS so that the decision makers can use the program with ease. This interactive DSS will include the graphical user interface, help facilities, versatility ability, uniformity, consistence, and informativeness.

- To develop a DSS to aid the decision makers to optimize and forecast the financial planning decision of SME.

4.2 Use Case Diagram

The use case diagram of this system is shown in figure 4.1. Six use cases are identified, namely, analysis balance sheet, analysis capital budgeting, analysis cash flow, analysis loan payment, analysis operating budgeting, and change financial model. The actors involved in this system are the financial controller and decision maker.

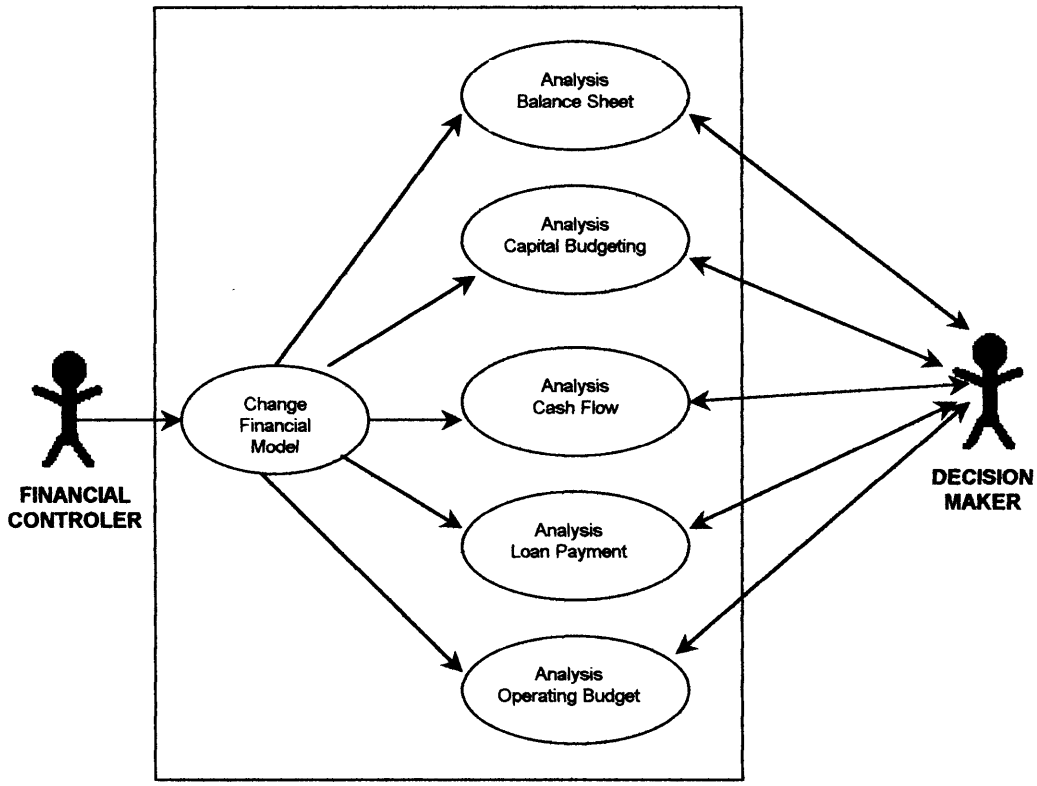


Figure 4.1: Use Case Diagram of SME Financial Planning DSS

4.2.1 Definition of Actor And Use Case

The actors and the use cases in this system are described in table 4.1

Actors	*Use Case*	*Description Of Use Case*
Decision Maker	Analysis Balance Sheet	The decision maker selects the balance sheet analysis. He or she may select the forecasting or goal seeking analysis to find a feasible solution.
	Analysis Capital Budgeting	The decision maker selects the capital budgeting analysis. He or she selects the what-if analysis to find a feasible solution.
	Analysis Cash Flow	The decision maker selects the cash flow analysis. He or she selects the what-if analysis to find a feasible solution.
	Analysis Loan Payment	The decision maker selects the loan payment analysis. He or she selects the what-if analysis to find a feasible solution.

	Analysis Operating Budget	The decision maker selects the operating budget analysis. He or she may select the forecasting or goal seeking analysis to find a feasible solution.
Financial Controller	Change Financial Model	The financial controller change the financial models use in the balance sheet analysis.
		The financial controller change the financial models use in the capital budgeting analysis.
		The financial controller change the financial models use in the cash flow analysis.
		The financial controller change the financial models use in the loan payment analysis.
		The financial controller change the financial models use in the operating budget analysis.

Table 4.1: Definition of Actors And Use Cases

4.3 System Requirement

The system requirement is discussed according to the system functionality and system feature.

4.3.1 System Functionality

The functionality of a system refers to the range of operational tasks it supports. The functions of this system are derived from what-if analysis, forecasting, and goal seeking.

What-if analysis allows the decision maker to manipulate the financial input data such as the loan interest rate in order to investigate the change against the existent solution.

Forecasting allows the decision maker to do the five years financial forecasting. The decision maker can manipulate the financial input data such as the inflation rate to forecast the future financial status.

Goal seeking is used to assist the decision maker to compute the amount of inputs necessary to achieve a desired level of an output (goal or target). It represents a backward solution approach (Turban and Aronson 1998: 58).

The functionality of the system is shown in table 4.2.

Requirement 1 The system is able to perform balance sheet analysis For balance sheet analysis, the key decision to support is the financial ratios analysis (liquidity ratios, debt management, and profitability ratios).
Requirement 2 The system is able to perform capital budgeting analysis. The key decision for capital budgeting analysis is to evaluate the capital invested.
Requirement 3 The system is able to perform cash flow analysis. The key decision for cash flow analysis is to do the what-if analysis towards the sales collections and net cash flow.
Requirement 4 The system is able to perform loan payment analysis. The key decision for loan payment analysis is to indicate the need of the loan to be made.
Requirement 5 The system is able to perform operating budget analysis. The key decision for operating budget analysis is to plan the daily operating activities.

Requirement 6 The system is able to allow changes to the financial model use in the balance sheet analysis.
Requirement 7 The system is able to allow changes to the financial model use in the capital budgeting analysis.
Requirement 8 The system is able to allow changes to the financial model use in the cash flow analysis.
Requirement 9 The system is able to allow changes to the financial model use in the loan payment analysis.
Requirement 10 The system is able to allow changes to the financial model use in the operating budget analysis.

Table 4.2: System Requirement In Term of Functionality

4.3.2 System Feature

The system feature is derived from the user characteristics of the system. Table 4.3 shows the feature of the system.

Requirement 1
User-friendly interface for decision makers.
Requirement 2
User-friendly interface for financial controller.

Table 4.3: System Requirement In Term of Feature

4.4 Summary

The use case model depicts the system requirements of the system. The system requirements are summarized as table 4.4.

Requirement 1
The system is able to perform balance sheet analysis.
Requirement 2
The system is able to perform capital budgeting analysis.
Requirement 3
The system is able to perform cash flow analysis.
Requirement 4
The system is able to perform loan payment analysis.
Requirement 5
The system is able to perform operating budget analysis.

Requirement 6 The system is able to allow changes to the financial model use in the balance sheet analysis.
Requirement 7 The system is able to allow changes to the financial model use in the capital budgeting analysis.
Requirement 8 The system is able to allow changes to the financial model use in the cash flow analysis.
Requirement 9 The system is able to allow changes to the financial model use in the loan payment analysis.
Requirement 10 The system is able to allow changes to the financial model use in the operating budget analysis.
Requirement 11 User-friendly interface for decision makers.
Requirement 12 User-friendly interface for financial controller.

Table 4.4: System Requirements Analysis

Chapter 5

SYSTEM ARCHITECTURE AND MODEL CONSTRUCTION

This chapter discusses about the architecture and design of the system. The system design consists of model-based subsystem and dialog subsystem. The model-based subsystem and dialog subsystem will be discussed in-depth in this chapter.

5.1 System Architecture

There are three subsystems management in the DSS conceptual architecture design as shown in figure 5.1. The subsystems are data subsystem management, model based subsystem management, and dialog subsystem management.

This system is an integration of the model based subsystem and dialog subsystem. For this reason, the data subsystem is not included in the construction of the design implementation. The processes of the system architecture for this project are described as below.

Step 1

During the data subsystem management, the external data source and internal data source supply the financial data and external financial indicator to the external database. This database provides the related financial input data to the data user interface. The decision maker can choose a financial statement in order to start the financial analysis. The decision maker can do the what-if analysis, forecasting or goal seeking to implement the financial analysis.

Step 2

After selecting the analysis function, the financial input data is sent to Microsoft Excel 2000 (snapshot database). The financial models are embedded in the snapshot database as the calculation engine of the analysis.

Step 3

The result of the analysis is sent to the decision maker through the model-user interface. Microsoft Visual Basic 6.0 has been used to develop the user interface.

Step 4

The decision maker can evaluate the output resulting from the analysis through the model-user interface. The process will end when the output result is feasible. Else, the decision maker can start the process loop again or

back to the analysis function. The decision maker can continue this process until he or she finds a feasible solution.

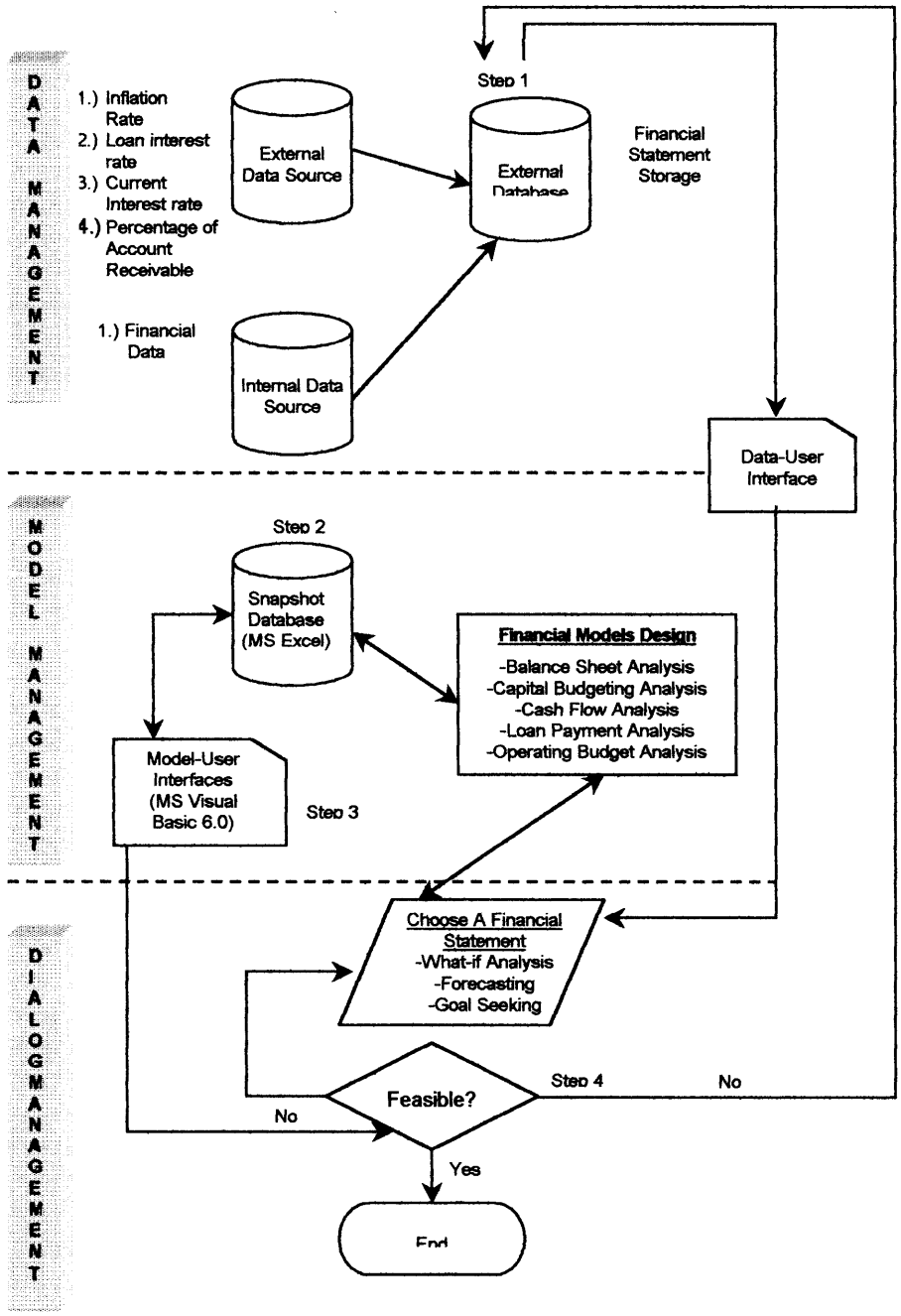


Figure 5.1: A Conceptual Architecture For SME Financial Planning DSS

5.2 Model Based Subsystem

Model management subsystem is a software package that includes financial, statistical, management science, or other quantitative model that provide the system's analytical capabilities, and appropriate software management. Modeling languages for building custom models are also included. This software is often called a model based management system (MBMS) (Turban and Aronson 1998: 79).

For this project Microsoft Excel 2000 is used to develop the financial models. The user is not aware of the existent of the financial models because the financial models are embedded in the spreadsheet as the calculation engine of this system.

The financial models are developed according to the five financial statements:

- Balance Sheet
- Capital Budgeting
- Cash Flow
- Loan Payment Analysis
- Operating Budget

5.2.1 Balance Sheet

Balance sheet is a statement of the firm's financial position at a specific point in time (Brigham 1995: 31). The balance sheet consists of asset, liability and equity.

Asset priority is ranked according to the liquidity of the asset. Liquidity means the length of time the asset takes to convert it to cash.

Below are the current asset priority ranking:

- The cash has the first priority in the asset ranking. Although the assets are all stated in terms of ringgit, only cash represents actual money;
- Accounts receivables (debtor) are bills that others owe the company (creditor);
- Inventories show the ringgit that the firm has invested in raw materials, work-in-process, and finished goods available for sales.

Other than current assets, there is fixed asset that the firm need to invest. Fixed assets consist of plant, equipment, vehicle, and etc. It depends on the necessity of the SME firm to invest in these assets. Mostly, the SME holder owns the

fixed asset. In this case, the asset is no more categorized as fixed asset but is owner equity of the SME.

Owner equity is either the asset or cash that invested by the SME holders in their business. In Malaysia, most of the SME holders supply capital (cash and assets) to the business or firm.

Liability is the obligation that the firm owes to the creditor (other firms that allow the credibility). It consists of the current (short-term) liabilities and long-term liabilities. Account payables, notes payables, and accruals are the items that categorized in the current liability.

Retained earnings is the portion that the firm's earnings that has been saved rather than paid out as dividends. As SMEs have only one owner (sole proprietorship), the retained earning will not be paid as the dividends. It is the portion of the firm's earnings that has been saved rather than invested in the firm. The earnings are saved for the future investment or emergency use.

5.2.1.1 Decision Process of Balance Sheet

As balance sheet consists of asset, liabilities, owner equity, and retained earnings, it can be used as the input data to the financial ratios analysis.

Financial ratios analysis is used to evaluate the financial status of a firm. In the balance sheet analysis, the SME decision maker has two alternatives: forecasting and goal seeking.

Forecasting provides liquidity analysis, debt management, and profitability ratio analysis. The SME decision maker can manipulate the financial indicator (e.g inflation rate) to evaluate the liquidity analysis, debt management, and profitability ratio analysis. The analysis process will end when the ratio analysis result is feasible. If not, the decision maker can go back to the forecasting alternatives. The decision maker can continue the process until he or she finds the feasible solution to support the decision.

However, goal seeking provides the alternatives: target inventory, target long-term debt, and target retained earnings analysis. The SME decision maker can seek the target item (inventory, long-term debt, or retained earning) he/she wants. The goal seeking process will end when the ratio analysis resulting from the target item is feasible. If not, the decision maker can go back to the goal seeking alternatives. The process loop will end until the decision maker finds a feasible solution to support the decision.

Figure 5.2 shows the decision process flow of the balance sheet analysis.

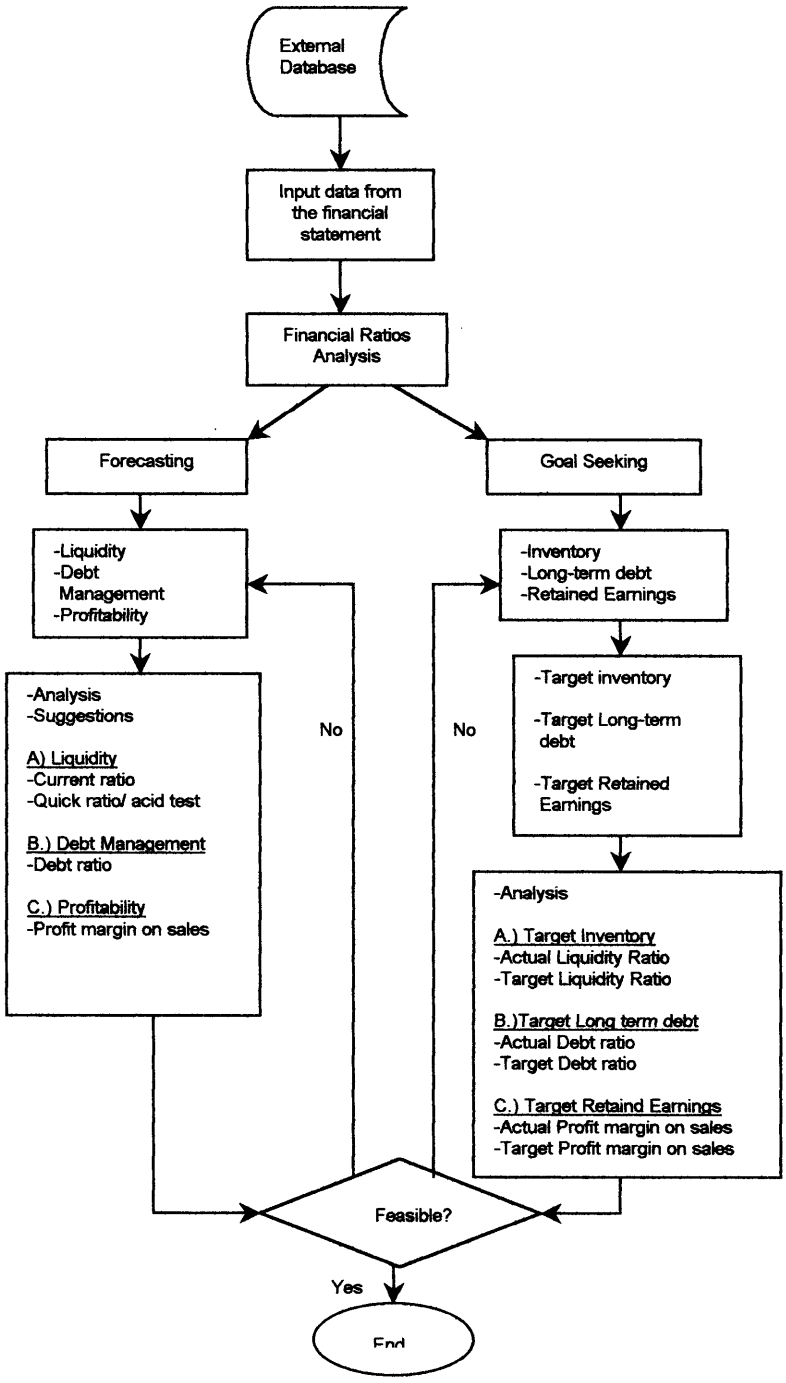


Figure 5.2 Decision Process of Balance Sheet

Forecasting

For the forecasting alternatives, there are liquidity analysis, debt management, and profitability. In SME firm, liquidity ratio is important to control the working capital. The working capital of the SME firm is small, therefore, the liquidity ratio analysis is needed to ensure the asset on hand is cash convertible. Inconvertible asset such as inventory can cause the firm facing inflexibility in cash turnover. This is dangerous when the firm needs cash to transact daily activities.

A debt management ratio is important for the SME firms to indicate the financial leverage (the use of debt financing). Since the capital of the SME firm is small, the SME owner can raise funds through debt. Creditors look to the equity, or owner-supplier funds, to provide a margin of safety. When the owner provides only a small proportion of the total financing, the risks of the enterprise are borne mainly by its creditors. In this case, the creditors especially the financial institution is reluctant to finance the firm. Hence the debt management is used to control the debt in the SME firm in order to get financial aid from the creditors.

Profitability ratios show the combined effect of liquidity, asset management, and debt management on operating results (Brigham 1995: 79). It is the net

result of a number of policies and decisions. For the SME financial planning needs, profit margin on sales (one of the profitability ratio) is needed to measure income per dollar of sales. Profit margin on sales is calculated by dividing net income by sales. The net income is assume to be equivalent with the retained earnings as SME firm do not have any stockholders. The total net income will be saved fully for the business use.

Goal Seeking

The goal seeking alternatives provide target inventory, target long-term debt, and target retained earnings analysis. Target inventory, target long-term debt, and target retained earnings analysis are important for the SME firm. These items may influence the financial ratios analysis.

The floatation of target inventory causes the liquidity ratio change. Therefore the SME decision maker needs to compare the actual and target liquidity ratio before the target inventory being achieved.

Other than that, the floatation of the target long-term debt also brings impacts to the debt ratio. For this reason, the SME decision maker needs to compare the actual and target debt ratio before the target long-term debt being achieved.

The change of the target retained earnings will also influence the profit margin on sales. Thus, the SME decision maker needs to compare the actual and target profit margin on sales before the target retained earnings being achieved

5.2.1.2 Financial Model of Balance Sheet

To evaluate the balance sheet analysis, there are several financial models that need to form. These financial models are formed according to the forecasting alternatives and goal seeking alternatives.

Financial Models For Forecasting

The financial model formulations are taken from Brigham (1995).

Notation: i =inflation rate

For the liquidity ratio, the current ratio and quick ratio or acid test is used as the indicator for the liquidity status of the SME firm.

The current ratio and quick ratio / acid test formulations for the SME firm are presented as follows:

Current Ratio =Current Assets / Current Liabilities (5.1)

Quick Ratio / Acid Test =

(Current Assets- Inventories) / Current Liabilities) (5.2)

Current Asset = (Cash + Account Receivables + Inventories) (5.3)

Current Liabilities = (Account Payables + Notes Payables + Accruals) (5.4)

When inflation rate change,

New Current Ratio =(Current Assets / Current Liabilities) * (1+i) (5.5)

New Quick Ratio / Acid Test

= [(Current Assets- Inventories) / Current Liabilities] * (1+i) (5.6)

To evaluate the liquidity ratio of the firm, industry ratio is needed. The industry ratio is the average ratio of the firms that appear in the same industry with the SME firm.

Comparison of the new liquidity ratio with the industry liquidity ratio will show the liquidity status of the firm in the industry. Higher ratio shows that the firm has more working capital to be invested. The creditors will have more interest in financing the higher liquidity ratio firms.

For the debt management, the debt ratio is used to measure the percentage of funds provided by the creditors to the SME firm.

The debt ratio formulations for the SME firm are presented as follows:

Debt Ratio = Total Debt / Total Asset (5.7)

Total Debt = (5.4) + Long-Term Debt (5.8)

Total Asset = (5.5) + Net Fixed Asset (5.9)

When inflation rate change,

New Debt Ratio =(5.7) * (1+i) (5.10)

The debt ratio of the firm can be evaluated by comparing it with the industry ratio.

Result of the comparison (new debt ratio with the industry debt ratio) is an indicator to attract creditor to finance the firm. Creditors prefer low debt ratios, because the lower ratio, the greater the cushion against creditors' looses in the event of liquidation (Brigham 1995: 77).

For the profitability ratio analysis, the profit margin on sales ratio is used to measures income per ringgit of the firm's sales.

The profit margin on sales ratio formulations for the SME firm are presented as follows:

Profit Margin On Sales Ratio = Retained Earnings / Sales (5.11)

When inflation rate change,

New Profit Margin On Sales Ratio = (5.11) * (1+i) (5 .12)

New Profit margin on sales ratio of the firm is compared to the industry profit margin on sales ratio in order to evaluate the income earned by the sales. Low profit margin on sales ratio indicates that the income generated by the sales is

low. In other words, the operating cost management is not sufficient as the operating cost of the firm is higher than the industry operating cost.

Financial Models For Goal Seeking

The financial models for goal seeking are taken from Brigham (1995).

For target inventory goal seeking, there is impact in the current ratio.

Here, the actual current ratio formulation is same as (5.1).

For the target current ratio formulation, the target inventory need to taken into consider.

Target Current Ratio = Target Current Asset / (5.4) (5.13)

Target Current Asset =

(Cash + Account Receivables + Target Inventories) (5.14)

The actual debt ratio formulation is same as (5.7).

The target long-term debt needs to be considered for the target debt ratio formulation.

Target Debt Ratio = Target Total Debt / (5.9) (5.15)

Target Total Debt =

Total Debt = (5.4) + Target Long-Term Debt (5.16)

The actual profit margin on sales ratio formulation is same as (5.11).

For the target profit margin on sales ratio formulation, the target retained earnings have to consider.

Target Profit Margin On Sales Ratio =

Target Retained Earnings / Sales (5.17)

5.2.2 Capital Budgeting

Capital budgeting is a many-sided activity that includes searching for new and more profitable investment proposal, investigating engineering and marketing considerations to predict the consequences of accepting the investment, and making economic analyses to determine the profit potential of each investment proposal (Bierman and Smidt 1988: 4).

The purpose of this capital budgeting subsystem is to help business management analyze the profit potential of the investment in plant, and equipment, marketing programs, research projects and the like.

In the typical investment decision management, decision maker makes a commitment of current resources in order to secure a stream of benefits in future years. However, there is no sharp conceptual difference between so-called capital and current expenditures; it is fair to say that all of the firm's expenditures are made in expectation of realizing future benefits (Levy and Sarnat 1990: 17).

The firm is continuously confronted by the problem of deciding if a proposed user of resources is worthwhile in terms of the prospective benefits. When the time horizon of the problem is relatively short, say less than one year (for example an increase in inventories or trade credit) both the costs and the benefits of a given proposal can be set out in current dollars. However, when a significant period of time elapses between the outlay and the benefits, the problem of evaluating and comparing costs and benefits becomes more difficult (Levy and Sarnat 1990: 18).

To evaluate the capital invested, the net present value (NPV), payback years period, minimum required interest rate, and the current interest rate are taken into considerations.

NPV is equal to the present value of future net cash flows, discounted at the marginal cost of capital (Brigham 1995: 337). The cost of capital is the current interest rate offered by the financial institutions.

Payback years period is the length of time required for the net revenues of an investment to cover the investment (Brigham 1995: 335).

Minimum required interest rate or internal rate of return (IRR) is the discount rate that forces the present value of a project's inflows to equal to the present value of its costs (Brigham 1995: 339).

5.2.2.1 Decision Process of Capital Budgeting

The decision process of capital budgeting starts from the external database (storage of the historical data of financial statements). The input data from the financial statements will be used in the capital budgeting.

The capital budgeting analysis focuses on the payback years period, index profitability, asset year last, NPV, and NPV graph.

The capital budgeting analysis will end when the analysis result is feasible. If not, the decision maker can continue the analyses by manipulating the input data. This process will continue until the decision maker finds a feasible solution.

Figure 5.3 shows the decision process flow of the capital budgeting analysis.

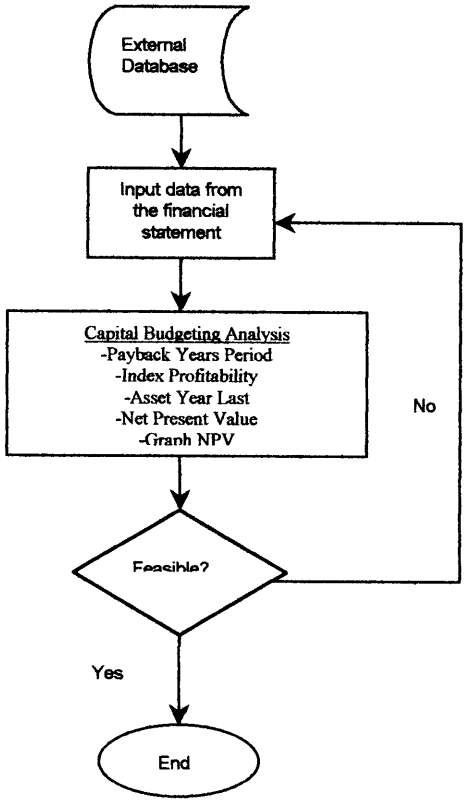


Figure 5.3: Decision Process of Capital Budgeting

5.2.2.2 Financial Model of Capital Budgeting

The financial models for capital budgeting are taken from Brigham (1995).

For the payback years period,

Payback Years Period = Asset Cost / [Asset Cost (IRR)] (5.18)

Where, Asset Cost = Capital Cost (use to invest / buy in assets,

marketing programs, research projects, and etc.)

For NPV,

$NPV = CF_0 + CF_1 / (1+k) + \ldots\ldots\ldots + CFn / (1+k)^n$

$$= \sum_{t=0}^{n} CFt / (1+k)t \qquad (5.19)$$

Where CF= cash inflow; k = current interest rate;

CF_t = expected net cash flow at period t

CF= Asset Cost * IRR (5.20)

For Index Profitability,

Index Profitability = PV of Cash Inflow (NPV) / Initial Investment

In this case,

Index Profitability = (5.19) / Initial Investment (5.21)

For Year Asset Last,

Year Asset Last = (5.18) + Year Asset Being Bought (5.22)

If the Index Profitability (5.21) is greater than 1, and the Payback Years Period (5.18) less or equal to 10 years, then the asset or project is reasonable to be invested. The greater the index profitability, the greater the asset or project is worth to be invested. Anyway, the payback years period shouldn't be more than 10 years as this may take longer time for SME to recover the investment. The cost of investment can be used for other investment where the payback years period is shorter.

5.2.3 Cash Flow

Cash flow is the actual net cash that a firm generates in some specified period (Brigham 1995: 39). This subsystem concentrates in the operating cash flow analysis.

The operating cash flow is the cash flow that arises from normal operations; the difference between sales revenues and cash expenses (Brigham 1995: 41). The input financial data of the operating cash flow are the sales forecast by the SME firm, price per unit of the product, cost per unit of the product, and cash outflows.

The cash outflows consist of total variable cost, rent, wages and salaries, insurance, and equipment payment. The sales forecast is the only resource of cash inflow of the firm. The sales forecast consists of account receivables that may not collect in time for the cash balance accounting period. Therefore, the cash inflows may be less than the sales forecast in the same time period. The balance of the sales forecast is carried to the next cash inflows for the following time period.

With the input financial data, the decision maker can perform the what-if analysis towards the sales collections and net cash flow. The analysis result

assists the decision maker to evaluate the cash balance in hand. The cash balance in hand is used for the operating activities or investment.

5.2.3.1 Decision Process of Cash Flow

The decision process begins from the external database where the input data from the financial statements is used in the cash flow analysis.

The output of the cash flow analysis focuses on the sales collections, net cash flow, and ending cash balances. The cash inflows analysis will end when the analysis result is feasible. If not, the decision maker can continue the process loop until he or she finds a feasible solution.

Figure 5.4 shows the decision process flow of cash flow analysis.

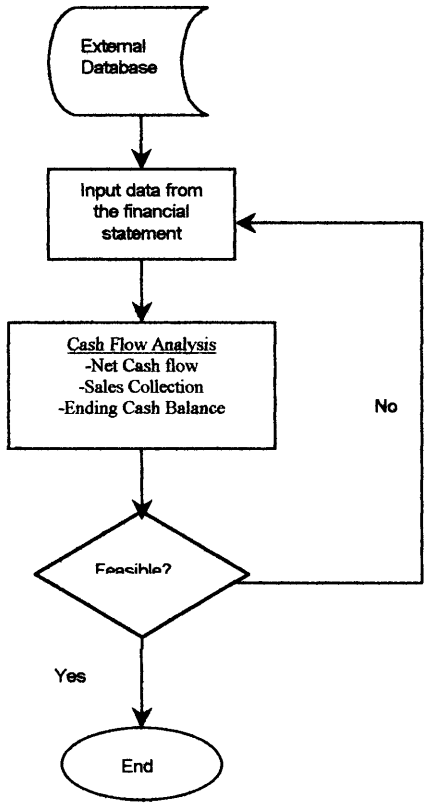


Figure 5.4: Decision Process of Cash Flow Analysis

5.2.3.2 Financial Model of Cash Flow

The financial models for cash flow are taken from Brigham (1995).

Notation: n = Percentage of the sales forecast to be account receivable

For sales collections,

Sales Collections ={[Sales Forecast * (1-n)] * Price Per Unit} (5.23)

For net cash inflows,

Net Cash Inflows =

[Cash Inflows or (5.23)] – Cash Outflows (5.24)

Cash Inflows =

(Total Variable Cost + Rent + Wages and Salaries +

Insurance + Equipment payment) (5.25)

For ending cash balance,

Ending Cash Balance = Net Cash Inflows + Starting Cash Balance (5.26)

Starting Cash Balance = Previous Ending Cash Balance (5.27)

5.2.4 Loan Payment Analysis

Loan Payment is the payment that needs to pay to the lender. The analysis is being made to indicate the need of the loan made.

For loan payment analysis subsystem, the amortized loan is used. The loan is repaid in an equal periodic amount (Brigham 1995: 224).

The present value of the total payment is calculated by discounting the loan made to the interest rate (offer by financial institution). Then, the present value of the total payment is discounted by the loan interest rate (base lending rate) to get the present value of the payment that needs to pay to the lender. The loan interest rate is the cost of the borrower. It is also the rate of return of the lender (Brigham 1995: 225).

5.2.4.1 Decision Process of Loan Payment Analysis

The decision process of loan payment analysis starts from the external database. From this database, the data of the financial statement is used as input to the loan payment analysis subsystem.

The decision maker can analyse the loan payment by having the output: present value of total payment, annual payment (after discounted by the loan interest rate), monthly payment (after discounted by the loan interest rate), and total number of monthly payment.

The analysis process will end when the decision maker find a feasible solution. If not, he or she can continue the process loop until he or she finds the feasible solution.

Figure 5.5 shows the decision process flow of loan payment analysis.

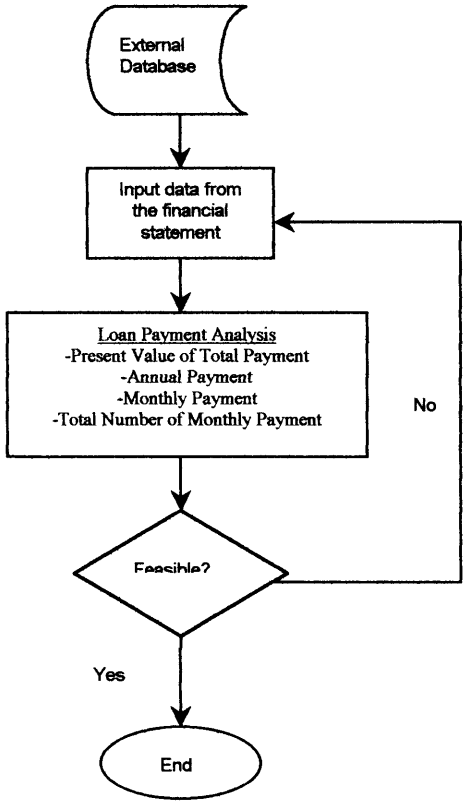


Figure 5.5: Decision Process of Loan Payment Analysis

5.2.4.2 Financial Model of Loan Payment Analysis

The financial models for loan payment analysis (amortized loans) are taken from Brigham (1995).

Notation: i = interest rate; t = year payment

For present value of total payment,

$$\text{Present Value Of Total Payment} = \text{Loan Made} / (1+i)^t \qquad (5.28)$$

For present value of annual payment,

Present Value Of Annual Payment =

$$\sum_{t=0}^{n}(5.28)*(1+LoanInterestRate)t \qquad (5.29)$$

For present value of monthly payment,

Present Value Of Monthly Payment = (5.29)/12 (5.30)

5.2.5 Operating Budget

Operating budget shows the daily business activities carry by the firm. The daily business activities budget is needed to plan the daily operating activities.

The operating budget analysis needs the unit sales, price of unit, operating variable cost, and operating fixed cost. The unit sales are the units of product that the firm can sell in a certain period (eg.3 months). And, the price of unit is the price that the products being sell in the market.

However, operating variable cost per unit is calculated by dividing the total variable cost to the quantity of the product made. The operating variable cost per unit includes material cost per unit, labor cost per unit, and transportation cost per unit.

The operating fixed cost is the total fixed cost of the firm. The operating fixed cost includes rent, payroll expenses, insurance, and equipment payment.

5.2.5.1 Decision Process of Operating Budget

As operating budget consists of unit sales, price of unit, operating variable cost, and operating fixed cost, it can be used as the input data for the forecasting and goal seeking alternatives.

In operating budget subsystem, forecasting provides report and graph analysis. This report shows the five-year operating budget forecasting statement (for 5 years). The graph analyses provide break-even sales versus margin of safety and revenue versus gross margin. The analysis process will end when the result of the report or graph is feasible. Else, the decision maker can manipulate the external financial indicator (e.g: inflation rate) until he or she achieves the feasible solution.

Whereas, goal seeking enables the decision maker to seek for the goal profit or target profit that need to be achieved. The indicator of the goal profit is the unit sales of the firm. The analysis process will end when the unit sales, resulting from the analysis are feasible. If not, the decision maker can continue the

process loop until he or she achieves a feasible solution needed. Figure 5.5 shows the decision process of operating budget.

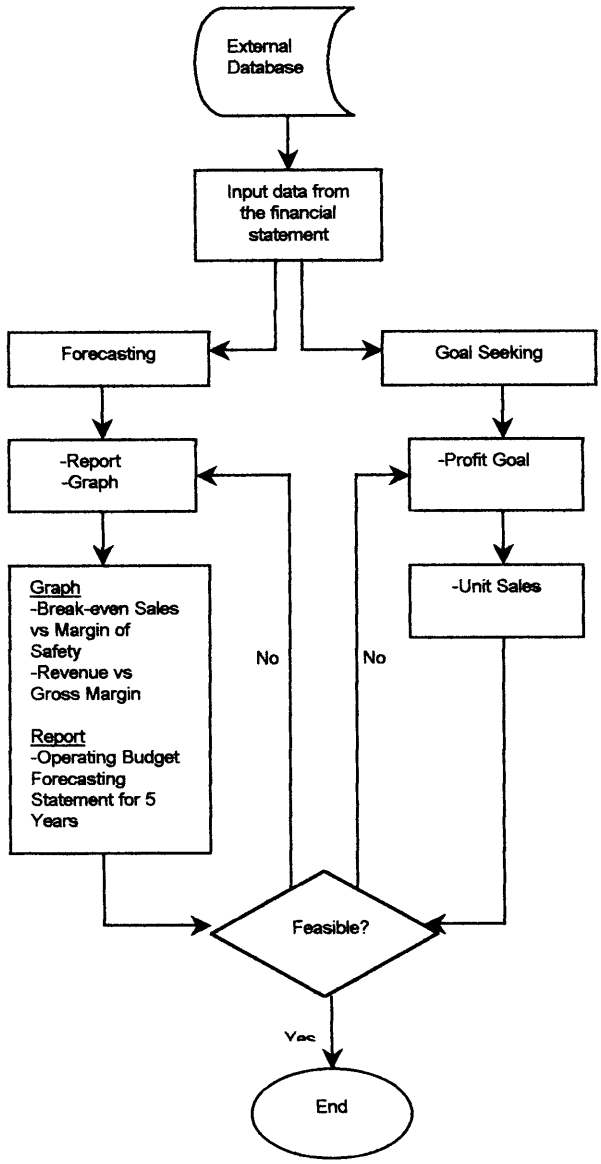


Figure 5.6: Decision Process of Operating Budget

5.2.5.2 Financial Model of Operating Budget

To evaluate the operating budget analysis, there are several financial models that need to form. These financial models are formed according to the forecasting alternatives and goal seeking alternative.

Financial Model For Forecasting

The financial models formulations are taken from Brigham (1995).

For break-even sales,

Break-even Sales =

Total Fixed Cost / Gross Margin Ratio (5.31)

Total Fixed Cost =

(Rent + Payroll Expenses + Insurance + Equipment Payment) (5.32)

Gross Margin Ratio

= (Gross Margin/ Revenue) * 100 (5.33)

For margin of safety,

Margin Of Safety =

[(Unit Sales-Break-Even Units) / Unit Sales]*100 (5.34)

Break-Even Units =

Break-Even Sales / Price Per Unit (5.35)

For revenue,

Revenue = Unit Sales * Price Per Unit (5.36)

For gross margin,

Gross Margin = (5.31) / Total Variable Cost Per Unit (5.37)

Financial Model For Goal Seeking

The financial model formulation for goal seeking is taken from Brigham (1995).

For the profit goal, the decision maker is interested in finding the unit sales need to sell in order to achieve the goal. For this reason, the unit sales are being calculated.

For target unit sales,

Target Unit Sales =

(Total Fixed Cost + Profit Goal) /

(Price Per Unit – Total Variable Cost Per Unit) (5.38)

5.2.6 SME Financial Planning DSS

SME Financial Planning DSS consists of balance sheet, capital budgeting, cash flow analysis, loan payment analysis, and operating budget. These financial analyses are needed to evaluate the financial status of the SME as well as to make future planning.

5.2.6.1 System Decision Process

The decision process of SME Financial Planning DSS begins from the external database, where the data of financial statement is stored. These financial data are the input of this system (SME Financial Planning DSS).

The decision maker has five alternatives: balance sheet analysis, capital budgeting analysis, cash flow analysis, loan payment analysis, and operating budget analysis.

After selecting the alternative, the decision maker may do what-if analysis in the capital budgeting analysis, loan payment analysis and cash flow analysis, or forecasting and goal seeking in balance sheet analysis and operating budget.

The decision maker can manipulate the input data in order to find a feasible solution. The analysis process will end when the solution is feasible. If not, the decision maker can continue the process loop until a feasible decision is provided.

Figure 5.7 shows the decision process flow of SME Financial Planning DSS

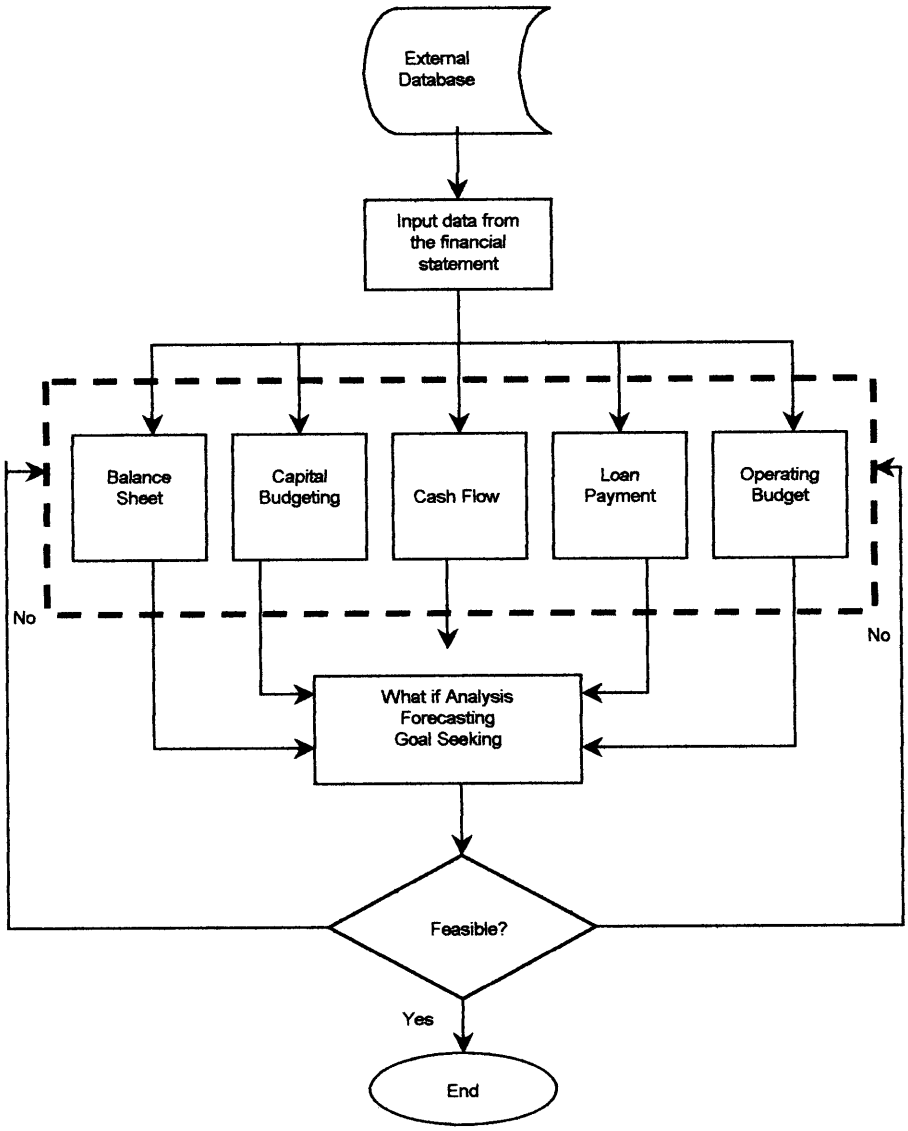


Figure 5.7: System Decision Process

5.3 Dialog Subsystem

The dialog subsystem covers all aspects of communication between a user and the DSS. These aspects include the factors that deal with the ease of use, accessibility, and human-machine interaction (Turban and Aronson 1998: 85).

The user interface component includes a natural language processor or standard objects. The standard objects such as pull-down menus and buttons can be built through graphical user interface (GUI) software.

The relationship between model based subsystem and dialog subsystem is shown in figure 5.8.

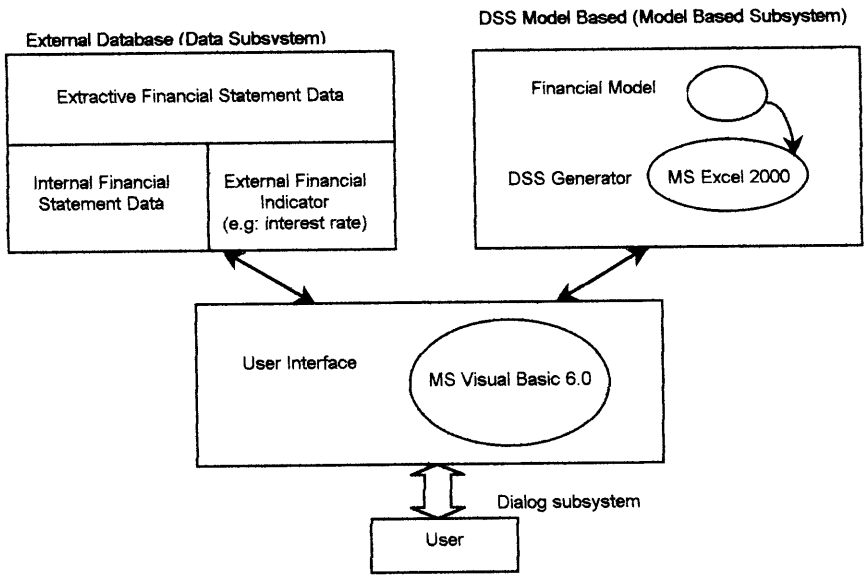


Figure 5.8: An Overview Design of SME Financial Planning DSS

5.4 Summary

This chapter describes about the architecture and design of the SME Financial Planning DSS. It focuses on the model-based subsystem and dialog subsystem.

The model-based subsystem uses the financial models in five financial statements, namely, balance sheet, capital budgeting, cash flow, loan payment, and operating budget. The system can perform the what-if analysis, forecasting, and goal seeking functions on the financial models.

The dialog subsystem addresses the communications between a SME Financial Planning DSS and users. The output of the analyses is presented in the form, text, graph, and report format. It is designed to be user-friendly to support the tasks of SME decision makers.

Chapter 6

TESTING AND EVALUATION

In this chapter, the testing and evaluation results are discussed. The system is tested and evaluated to verify that it does meet the stated functionality and feature.

6.1 System Functionality

The system is tested in terms of functions capabilities. Table 6.1 shows the functionality that meets the system requirements.

System Requirement	Testing And Evaluation Result
Requirement 1 System is able to perform balance sheet analysis	• ***What-if Analysis:*** The user can manipulate the financial input data (asset value, liability value, owner equity, and retained earning) to investigate the change in the existent financial status. • ***Forecasting:*** The user can forecast the liquidity ratio, debt ratio, and profitability

	ratio by manipulating the inflation rate. • ***Goal Seeking:*** The user can compute the liquidity ratios, debt ratio, and profitability ratio that necessary to achieve the target inventory, target long-term debt, and target retained earning.
Requirement 2 System is able to perform capital budgeting analysis.	• ***What-if Analysis:*** The user can manipulate the financial input data (asset cost, current interests rate and minimum required interest rate) to investigate the change in the payback period and index profitability of the asset or capital invested. • ***Forecasting:*** The user can forecast the payment period and net present value of the capital or asset invested by manipulating the interest rate and minimum required interest rate.

Requirement 3 System is able to perform cash flow analysis.	***What-if Analysis:*** The user can manipulate the financial input data (percentage of the sales forecast to be account receivable, sales forecast, and cash inflows elements) to investigate the change in the cash flow.
Requirement 4 System is able to perform loan payment analysis.	• ***What-if Analysis:*** The user can manipulate the financial input data (current interest rate, total year payment being made, and loan interest rate) to investigate the change in the total loan made.
Requirement 5 System is able to perform operating budget analysis.	• ***What-if Analysis:*** The user can manipulate the financial input data (price per unit sale, unit sales, variable cost and fixed cost) to investigate the change in the operating budget analysis. • ***Forecasting:*** The user can do the five-year forecasting in this subsystem. The subsystem enable the user to forecast the break-even sales, margin of safety, revenue, and gross

	margin, operating budgeting statement by manipulating the inflation rate and rate of unit sales changes. • ***Goal Seeking:*** The user can compute the unit sales that necessary to achieve the target profit.
Requirement 6 The system is able to allow changes to the financial model use in the balance sheet analysis.	• The financial controller can change the financial models in the balance sheet through Microsoft Excel 2000 (spreadsheet).
Requirement 7 The system is able to allow changes to the financial model use in the capital budgeting analysis.	• The financial controller can change the financial models in the capital budgeting through Microsoft Excel 2000 (spreadsheet).
Requirement 8 The system is able to allow changes to the	• The financial controller can change the financial models in the cash flow through

financial model use in the cash flow analysis.	Microsoft Excel 2000 (spreadsheet).
Requirement 9 The system is able to allow changes to the financial model use in the loan payment analysis.	• The financial controller can change the financial models in the loan payment analysis through Microsoft Excel 2000 (spreadsheet).
Requirement 10 The system is able to allow changes to the financial model use in the operating budget analysis.	• The financial controller can change the financial models in the operating budget through Microsoft Excel 2000 (spreadsheet).

Table 6.1: System Functionality That Meets The System Requirements

6.2 System Feature

The system has been tested in the academic environment (Universiti Utara Malaysia). Therefore, the users are the management and financial students in the campus. For this reason, the system has fulfilled the user interface characteristics.

However, the selected sample users have financial background and this is different compared to the SME decision makers, who do not have financial background.

The system feature is tested in terms of time length taken by the users and error rate.

- **The Length Of Time**

The variance of the length of time taken by the sample users should be as small as possible. DeSanctics et. al (1994) suggested a guideline of 1.5 minutes to complete each operational (or feature) task in end user software.

- **Error Rate**

According to DeSantics et. al (1994), designer would expect error-free completion of the operational tasks, but acceptable error rate perhaps stated as number of errors per minute can be established as a benchmark.

Table 6.2 shows the feature that meets the system requirements.

System Requirement	Testing And Evaluation Result
Requirement 1 User friendly interface for decision makers	• ***Time Length:*** The sample users have used not more than three minutes to complete two operational tasks. • ***Error Rate:*** The sample users perform a low error rate when performing the operational tasks.
Requirement 2 User friendly interface for financial controller	• ***Time Length:*** The sample users have used the spreadsheet to change the financial models. • ***Error Rate:*** The sample users who have financial knowledge perform a low error rate to change the financial models.

Table 6.2: System Feature That Meets The System Requirements

6.3 Summary

The system functionality and feature are tested in the academic environment (University Utara Malaysia). From the testing, the system fulfills the functionality requirement 1, 2, 3, 4, and 5, which meets the end user requirements. However, the functionality requirement 6, 7, 8, 9, and 10 were not achieved because the system requires the administrator (financial controller) to change the financial models using spreadsheet. This means that

the financial controller needs to have financial and spreadsheet knowledge in building the financial models. For this reason, the system administration function has to be improved so that it is more user-friendly.

For the feature requirements, requirement 1 and 2 were partly fulfilled as the system has a lower response time. This could be improved by having a higher performance PC.

Table 6.3 summarized the testing and evaluation results. As a conclusion, the system has fulfilled the end users' (decision makers) requirements (Requirement 1, 2, 3, 4, 5,and 11). However, the system must be made more user-friendly for the system administrators (Requirement 6, 7, 8, 9, 10, and 12).

Requirement	What-If Analysis	Forecasting	Goal Seeking	Spread -sheet based	Time Length	Error Rate
Requirement 1	✓	✓	✓			
Requirement 2	✓	✓				
Requirement 3	✓					
Requirement 4	✓					
Requirement 5	✓	✓	✓			
Requirement 6				✓		
Requirement 7				✓		
Requirement 8				✓		
Requirement 9				✓		
Requirement 10				✓		
Requirement 11					✓	✓
Requirement 12					✓	✓

Table 6.3: Testing And Evaluation Overall Result

Chapter 7

CONCLUSION AND RECOMMENDATION

This chapter reviews the overall progression of the project. The contributions and limitations of the project will be highlighted in this chapter. Following this, brief recommendations will be given as contributions to the future development.

7.1 Contributions Of The Project

The major contribution of this project is the development of a low cost and model based SME end user oriented financial planning DSS.

The integration of Microsoft Excel 2000 (spreadsheet based) and Microsoft Visual Basic 6.0 user interface has successfully produced a cheaper financial planning DSS. It will be more affordable for the SME to acquire compared to available special-purpose language softwares in the market such as Visual IFPS / Plus, ENCORE Plus!, Comshare Decision, DecisionPro, Pilot Software, and Analytical Engine.

The calculation engine of this DSS is built by generating the financial models into the Microsoft Excel spreadsheets. The form of the financial models is a contribution to the model based subsystem. This is because the financial models are reusable to other financial related DSS.

7.2 Limitations Of The Project

The project consists of several limitations. Below are the limitations that arise throughout the development process:

- The system developed in this project is a prototype system. As such, it does not have characteristics of a full function DSS.
- The prototype system does not ignore the existent of Microsoft Excel 2000. The users (SME decision makers) are aware of the existent of Microsoft Excel 2000.
- The system is a calculation or analysis engine for the SME decision makers, as it does not support the historical data storage. The historical data is taken from the external database.
- The system is not a group decision support system, which means it is used for standalone-company such as a sole-proprietary company that does not have any subsidiary.

7.3 Recommendations Of The Project

To overcome the limitations mentioned, several suggestions have been recommended. The suggestions are:

- The prototype system can be extended to a full functional system.
- The developer can develop a database that stores the financial historical data. The database can be developed with Microsoft Access 2000, Microsoft SQL Server, or the like.
- This system can be extended to group DSS (GDSS) as the SME may have subsidiaries. The group DSS (GDSS) can be developed by using web based technology. This can be implemented by integrating the DSS with any database in a CGI (Common Gateway Interface) standard.

7.4 Summary

The major contribution of this project is the development of a low cost and model based SME end user oriented financial planning DSS. However, there are several limitations to this system in terms of expansion. Several suggestions have been recommended to overcome the limitations, such as the system can be connected to a financial historical database or extended to a full functional system or GDSS. These recommendations may be useful for the future DSS development.

Currently, the system is tested in an academic environment where the selected sample users have a fairly good financial background. It is recommended that the future work need to be done on testing the system applicability in the actual SME environment.

Bibliography

Bennett, Simon et. al (1999). *Object Oriented Systems Analysis And Design Using UML.* England: McGraw-Hill Companies.

Bierman, Harold & Smidt, Jr., Seymour(1988). *The Capital Budgeting Decision: Economic Analysis Of Investment Projects.* 7th ed. New York: Macmillan Publishing Company.

Brigham, Eugene F. (1995). *Fundamentals Of Financial Management.* 7th ed. Florida: Dryden Press.

Chee Peng, Lim (1986). *Small Industry In Malaysia.* Kuala Lumpur: Berita Publishing Sdn. Bhd.

Comshare Inc.2000 (2000). Can you mine the information gold in your OLAP database? [www document].URL. http://www.comshare.com

DeSanctis, Gerardine (1994). "The Meaning Of The Interface: Functional And Holistic Evaluation Of A Meeting Software System". *Decision Support System* 11: 319-335.

Ferox Microsystems, Inc. (2000). ENCORE! Product Overview. [www document]. URL.http://www.ferox.com.

Holsapple, Clyde W. & Whinston, Andrew B. (1996). *Decision Support Systems: A Knowledge-Based Approach.* United States Of America: Course Technology.

Hokey Min (1989). "A Model-Based Decision Support System For Locating Banks". *Information & Management* 17:207-215.

Jones V. Christopher (1995). " Developments In Graph-Based Modeling For Decision Support". *Decision Support Systems* 61-174.

Kim, Chung Sook (1987). *A Formal Structure For The Evaluation Of Decision Support System Generators (DSSG): The Systems Approach.* Dissertation Information Service. Texas Tech University.

Knight Murphy & Associates Pty. Ltd. (2000). Like More POP4WIN Information? [www.document]. URL. http:// www.ozemail.com.au/~tknight/pop.hmtl

Langen Dieter (1989). "An (Interactive) Decision Support System For Bank Asset Liability Management". *Decision Support Systems* 5: 389-401.

Levy, Haim & Sarnat, Marshall (1990). *Capital Investment & Financial Decisions.* 4th ed. University Press, Cambridge: Prentice Hall Inc.

Lumina Decision Systems. (2000). Who's Doing What With Analytica. [www.document].URL. http://www.lumina.com/solutions/financial_application.html

Pilot Software (2000). Customer Support Overview. [www document]. URL. http://www.pilotsw.com/services/cust_support.html

Sankar, Chetan S. et. al (1995). "A DSS User Interface Model to Provide Consistency And Adaptability". *Decision Support Systems* 13: 93-104.

Sen, Arun & Vinze, Ajay (1996). "Introduction: DSS On Model Formulation". *Decision Support Systems* 17: 239-240.

Siskos, Y. et. al (1994). "An Integrated DSS For Financing Firms By An Industrial Development Bank In Greece". *Decision Support Systems* 12: 151-168.

Srivivasan, Venkat & Kim H. Yong (1986). "Decision Support For Integrated Cash Management". *Decision Support Systems* 2: 347-363.

Turban, Efraim & Aronson, Jay E. (1998). *Decision Support Systems And Intelligent Systems.* 5th ed. United States 0f America: Prentice Hall Inc.

Van Horne, James C. (1995). *Financial Management And Policy.* 10th ed. New Jersey: Prentice Hall Inc.

Appendix A User Manual For SME Financial Planning DSS

1.0 System Requirements

The following are the software and hardware requirements to support this system.

i.) Software

- Operating System Windows® 95 or Windows® 98
- Microsoft Excel 2000
- Microsoft Visual Basic 6.0

ii.) Hardware

- A Pentium-based PC-compatible computer system
- 32 MB of RAM (64 MB recommended)
- 400 MB of disk space
- An SVGA-compatible display (16bit or more colors recommended)

2.0 System Installation

There are two softwares that need to be installed before this system can be run. These softwares are Microsoft Excel 2000 and Microsoft Visual Basic 6.0.

2.1 The Installation Of Microsoft Excel 2000

Step 1: Install Microsoft Excel 2000 into CD ROM.

Step 2: Click Window Explorer icon.

Step 3: Choose drive C.

Step 4: Choose the setup icon and follow the setup instructions.

2.2 The Installation Of Microsoft Visual Basic 6.0

Step 1: Install Microsoft Visual Basic 6.0 into CD ROM.

Step 2: Click Window Explorer icon.

Step 3: Choose drive C.

Step 4: Choose the setup icon and follow the setup instructions.

3.0 How To Start The System

Step 1: Insert CD into CD ROM

Step 2: Click Window Explorer icon.

Step 3: Choose drive D (CR ROM drive).

Step 4: Click SMEFinancial PlanningDSS program

4.0 User Guides

The system begins with the front page shown in figure 1. To continue the system, click ***OK*** button.

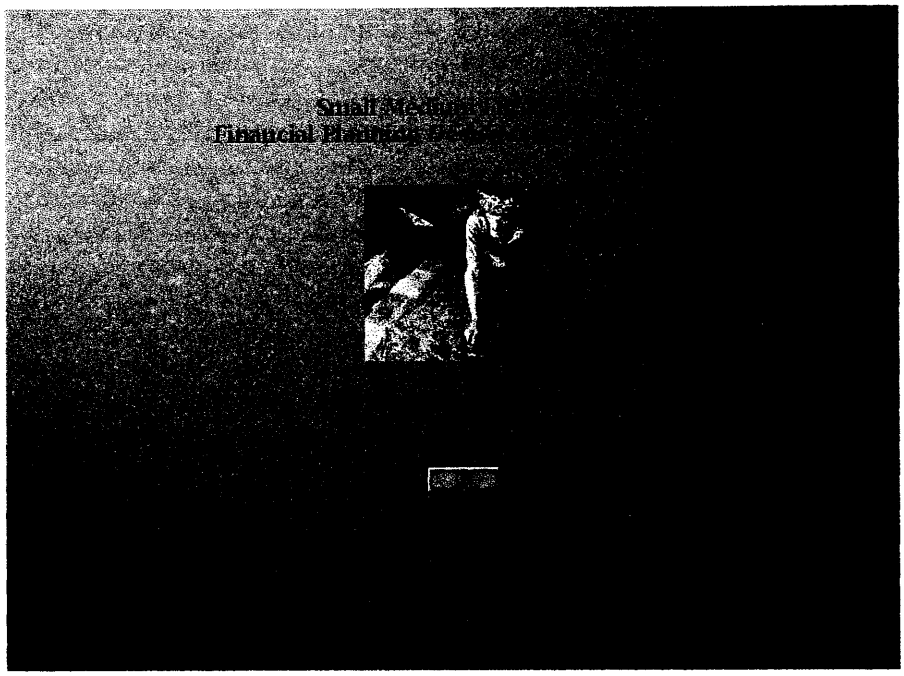


Figure 1: System Front Page

Then the system will ask for password to identify the authorized person before the user start to use the system. Figure 2 shows the login password window.

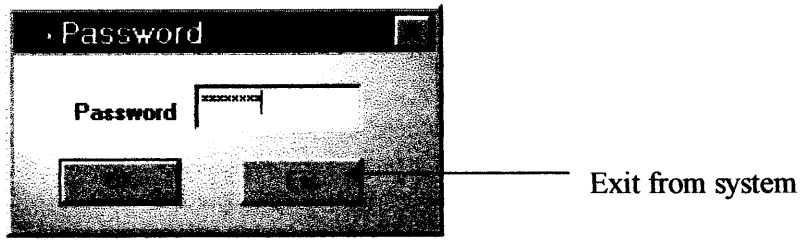


Figure 2: Login Password Window

If you are the authority person, the main menu will be displayed on the screen as figure 3.

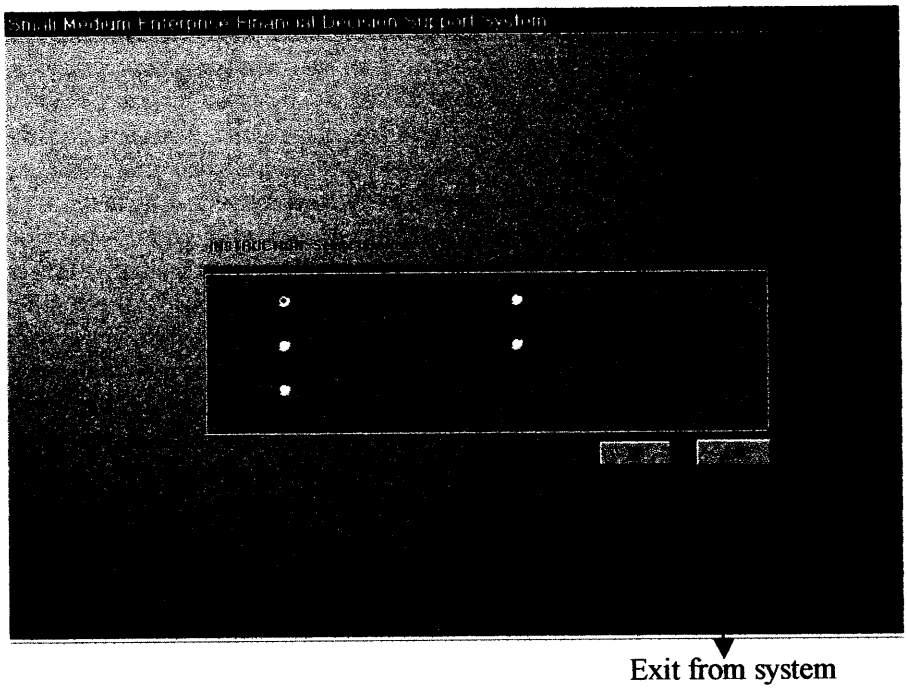


Figure 3: Main Menu

The main menu consists of five subsystems:

- Capital Budgeting
- Loan Payment Analysis
- Operating Budget
- Balance Sheet Analysis
- Cash Flow

4.1 Capital Budgeting

When you click the ***Capital Budgeting*** radio button, the figure 4 will be displayed.

This form enables the decision maker to manipulate the input data to find the feasible solution to support the decision.

This form provides the output:

- Graph NPV (net present value) for the asset
- Payback years period for the asset
- Index Profitability of the asset
- Asset last for the asset (to recover the cost)

Function of the buttons:

To return the main menu, user can click ***Menu*** button.

To analysis the result, user can click ***Analysis*** button.

To seek for help, user can click ***Help*** button.

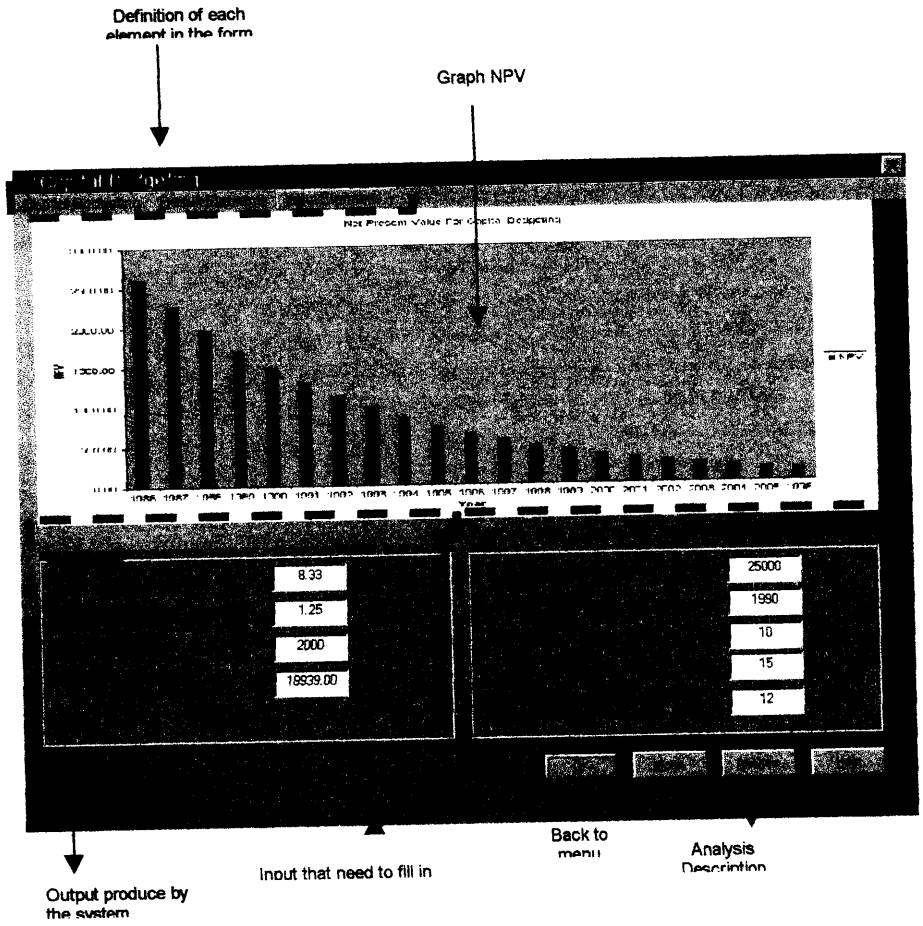


Figure 4: Capital Budgeting

4.2 Loan Payment Analysis

When you click the ***Loan Payment Analysis*** radio button, the figure 5 will be displayed.

This form enables the decision maker to manipulate the input data to find the feasible solution to support the decision.

This form provides the output:

- Present Value Of Total Payment
- Annual Payment
- Monthly Payment
- Total Number Of Monthly Payment

Function of the buttons:

To return the main menu, user can click ***Menu*** button.

To seek for help, user can click ***Help*** button.

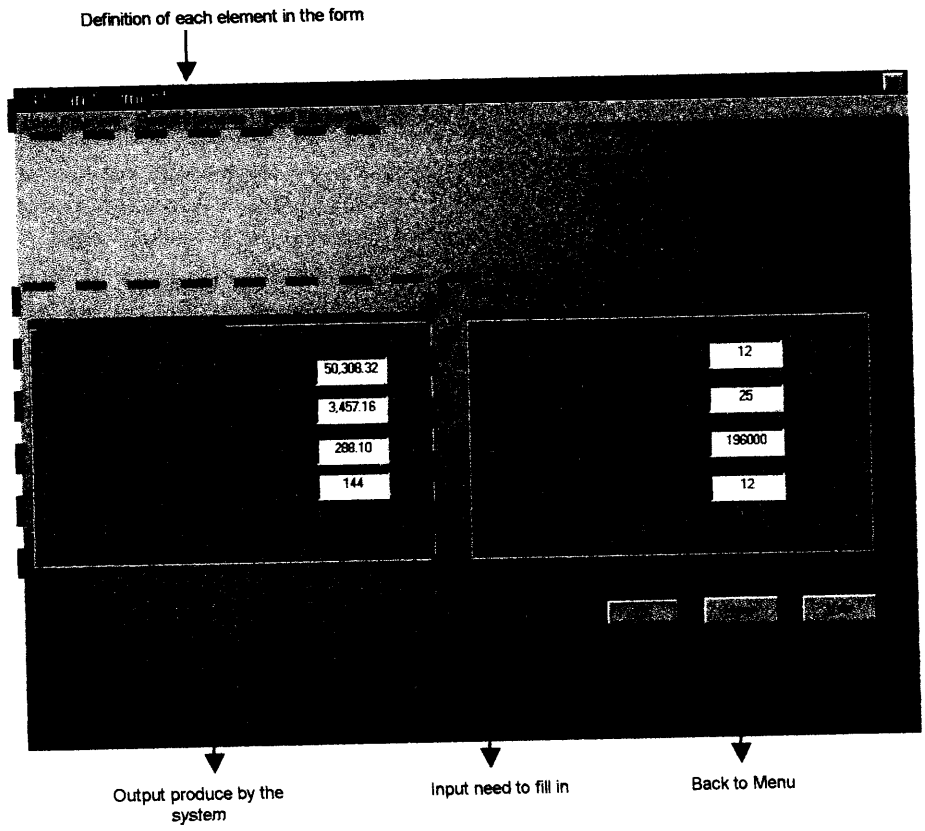


Figure 5: Loan Payment Analysis

4.3 Operating Budget

When you click the ***Operating Budget*** radio button, the figure 5 will be displayed.

This form enables the decision maker to manipulate the input data to find the feasible solution to support the decision.

Function of the buttons:

To return the main menu, user can click ***Menu*** button.

To seek for help, user can click ***Help*** button.

To continue the system, user can click ***OK*** button.

When the user click ***OK*** button, figure 6 will be displayed.

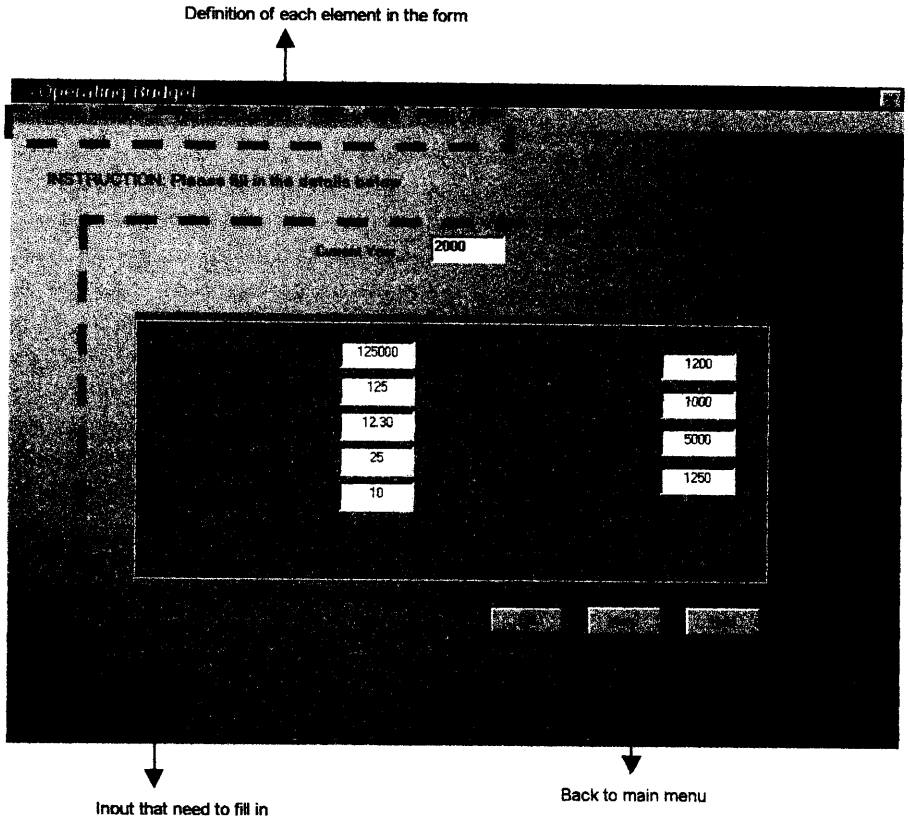


Figure 5: Input Form For Operating Budget

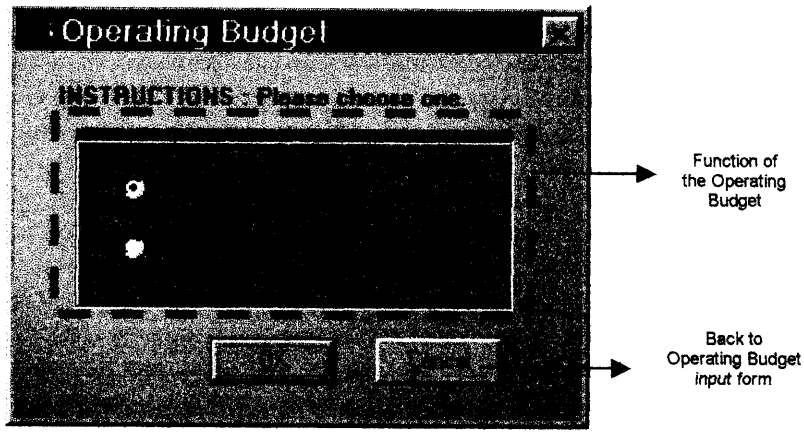


Figure 6: Function Menu Of Operating Budget

4.3.1 Forecasting

When you click the ***Forecasting*** radio button, the figure 7 will be displayed.

User can choose:

- Report
- Graph

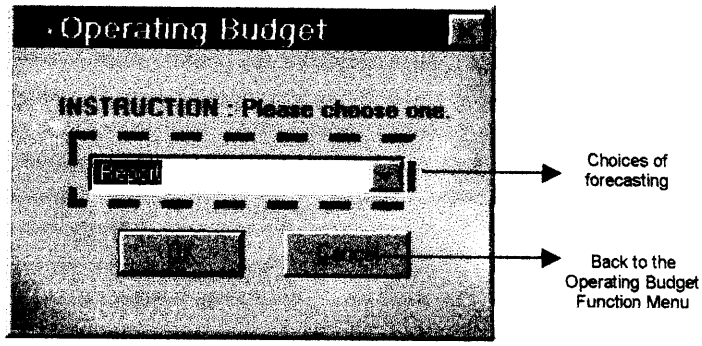


Figure 7: Forecasting Menu

4.3.1.1 Report

When user selects ***Report***, figure 8 will be displayed.

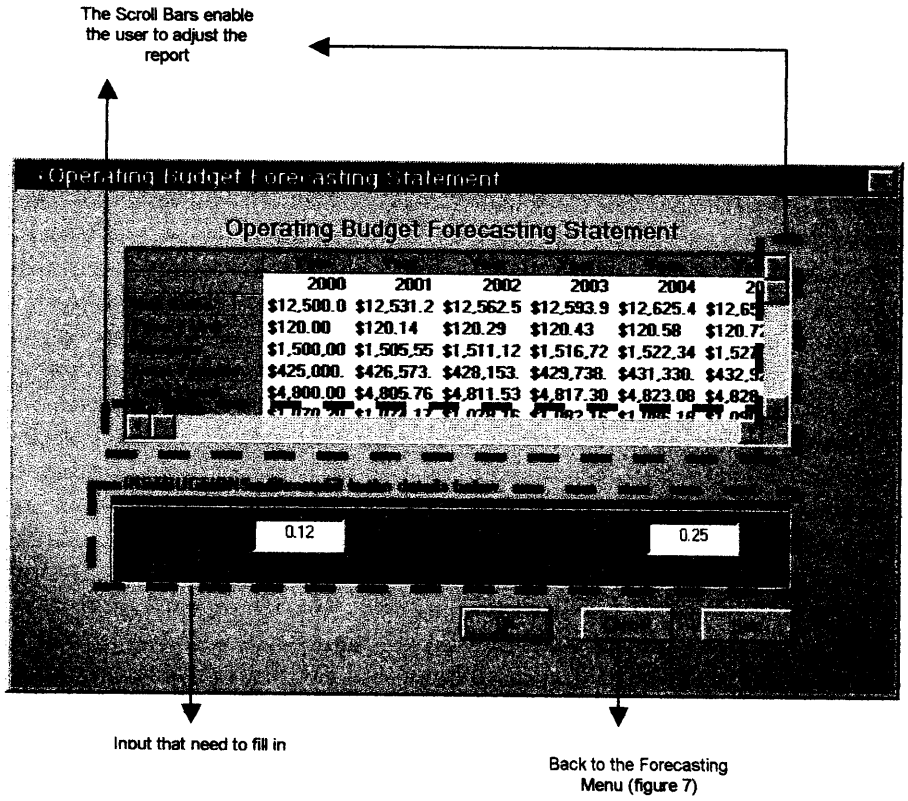


Figure 8: Report

This form enables the decision maker to manipulate the input data (Inflation Rate and / or Rate Of Unit Sales Changes) to find the feasible solution to support the decision.

Function of the buttons:

To seek for help, user can click ***Help*** button.

4.3.1.2 Graph

When user selects ***Graph***, figure 9 will be displayed.

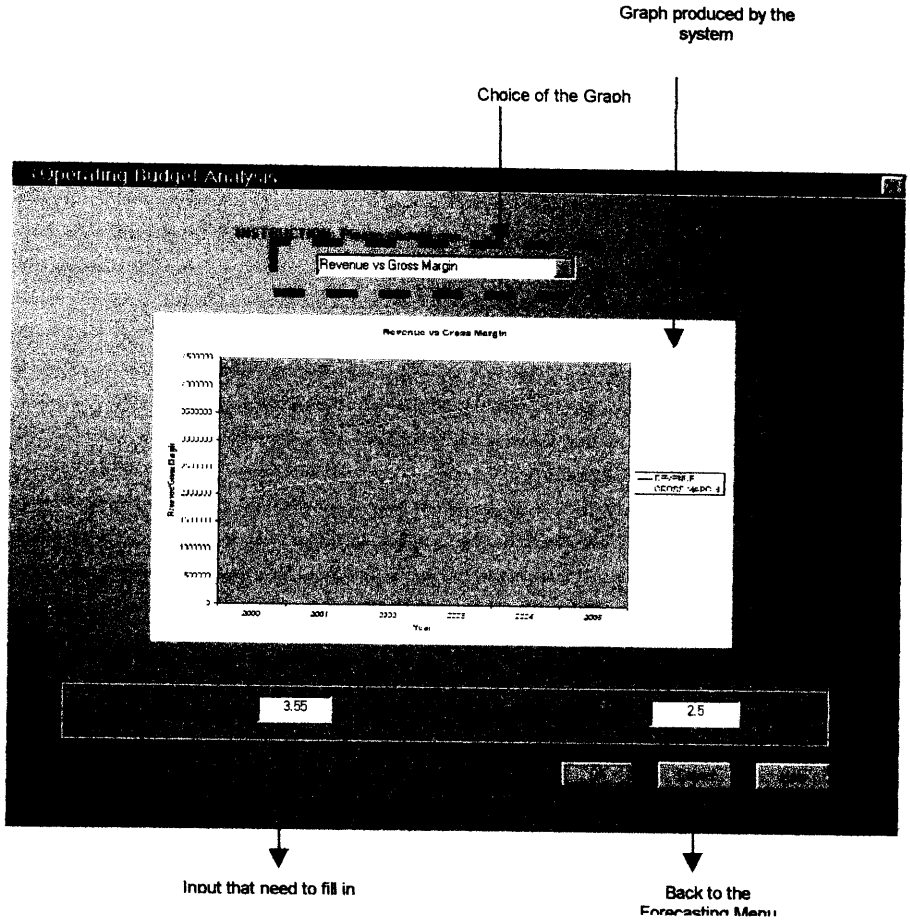


Figure 9: Graph

This form enables the decision maker to manipulate the input data (Inflation Rate and / or Rate Of Unit Sales Changes) to find the feasible solution to support the decision.

Function of the buttons:

To seek for help, user can click ***Help*** button.

4.3.2 Goal Seeking

When you click the ***Goal Seeking*** radio button, the figure 10 will be displayed.

User can choose:

- Year for the goal profit

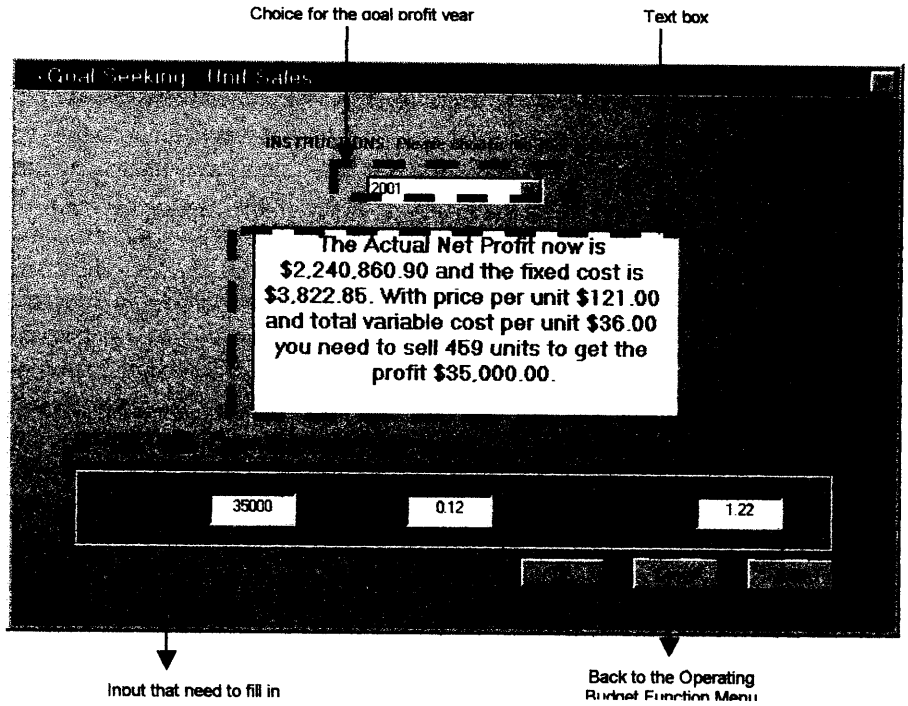


Figure 10: Goal Seeking

This form enables the decision maker to manipulate the input data (Inflation Rate and / or Rate Of Unit Sales Changes) to find the feasible solution to support the decision.

The output of goal seeking will be shown in the text box:

- Actual Net Profit
- Fixed Cost
- Variable Cost
- Price Per Unit
- Unit Sales Need To Achieve The Goal Profit

Function of the buttons:

To seek for help, user can click ***Help*** button.

4.4 Balance Sheet Analysis

When you click the ***Balance Sheet Analysis*** radio button, the figure 11 will be displayed.

This form enables the decision maker to manipulate the input data to find the feasible solution to support the decision.

Function of the buttons:

To return the main menu, user can click ***Menu*** button.

To seek for help, user can click ***Help*** button.

To continue the system, user can click ***OK*** button.

When the user click ***OK*** button, figure 12 will be displayed.

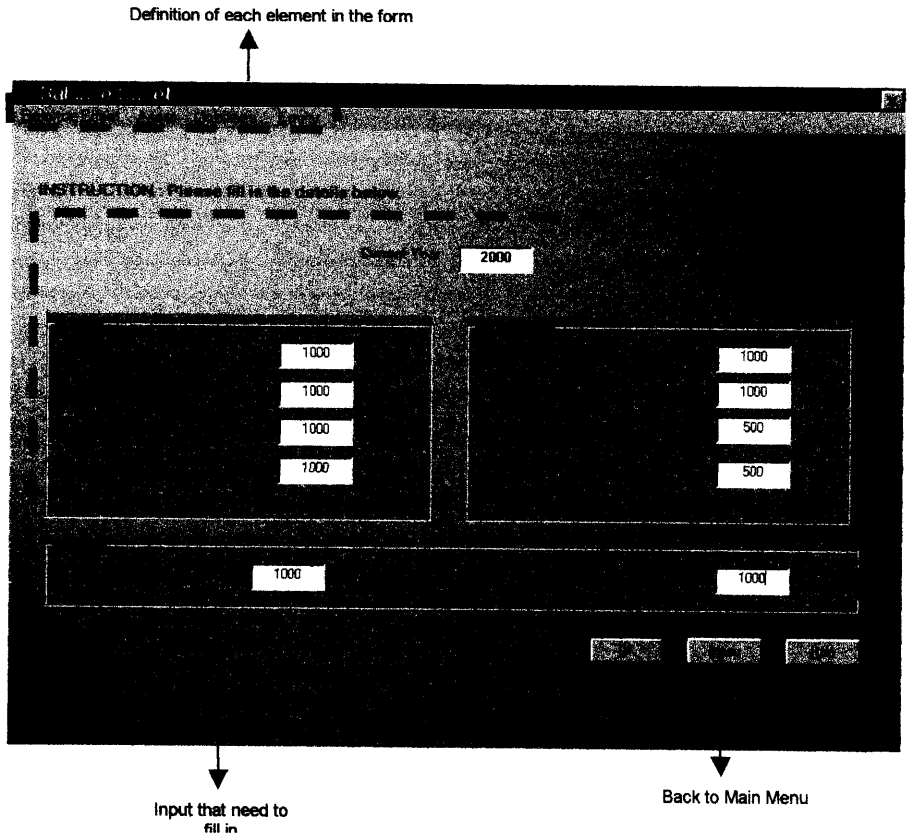


Figure 11: Input Form For Balance Sheet Analysis

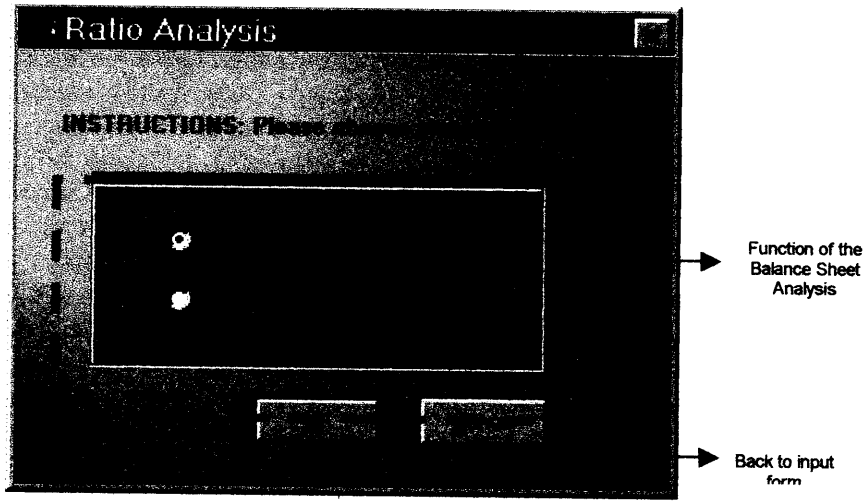


Figure 12: Function Menu For Balance Sheet Analysis

4.4.1 Forecasting

When you click the ***Forecasting*** radio button, the figure 13 will be displayed.

User can choose:

- Liquidity Ratio
- Debt Management (under constructions)
- Profitability Ratio (under constructions)

This form enables the decision maker to manipulate the input data (inflation rate) to find the feasible solution to support the decision.

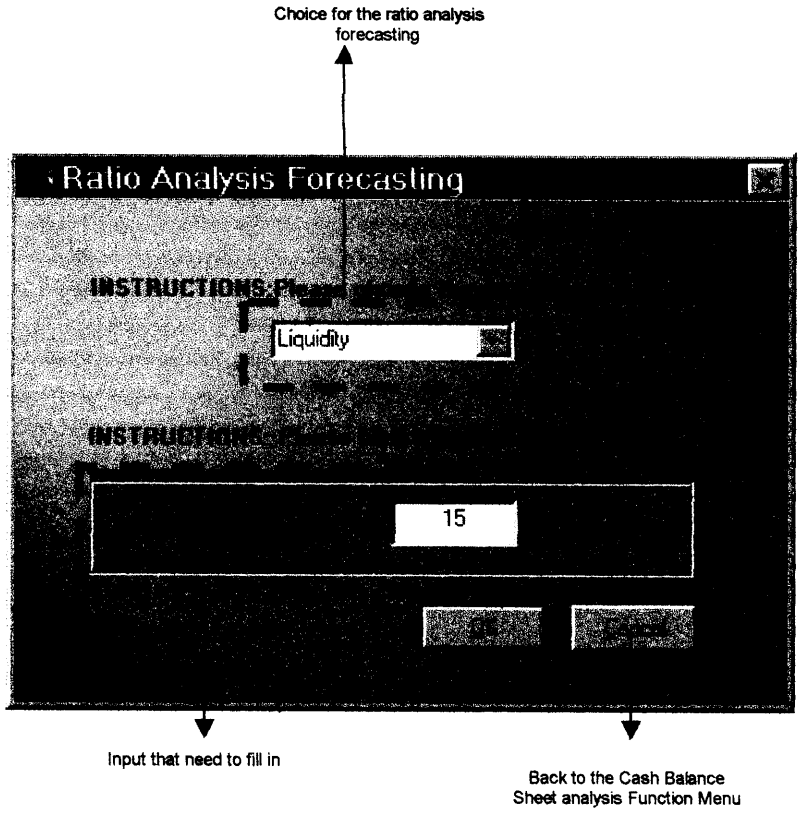


Figure 13: Ratio Analysis Forecasting

For this Balance Sheet Analysis forecasting function, only the liquidity analysis user manual will be shown. Others are still under constructions.

When user click ***OK*** button, figure 14 will be shown.

4.4.1.1 Output Of Forecasting

This form enables the decision maker to compare the average liquidity ratios (industry) to the liquidity ratios of the firm.

Function of the buttons:

To seek for help, user can click ***Help*** button.

The output of the forecasting function:

- Result of Liquidity Ratios of the firm
- Liquidity analysis
- Suggestion for the analysis result

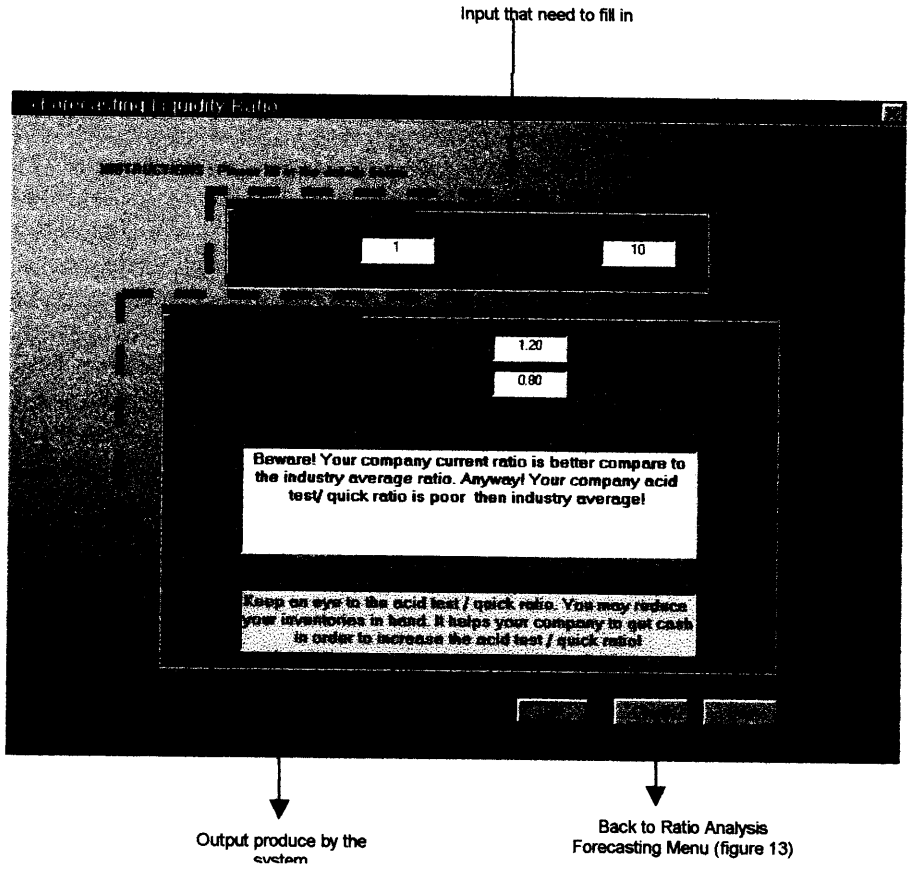


Figure 14: Output Of Ratio Analysis Forecasting

4.4.2 Goal Seeking

When you click the ***Goal Seeking*** radio button, the figure 15 will be displayed.

User can choose:

- Year of the target item and
- Target Inventory or
- Target Long-Term Debt (under constructions) or
- Target Retained Earning (under constructions).

For this Balance Sheet Analysis goal seeking function, only the target inventory user manual will be shown. Others are still under constructions.

When user click ***OK*** button, figure 14 will be shown.

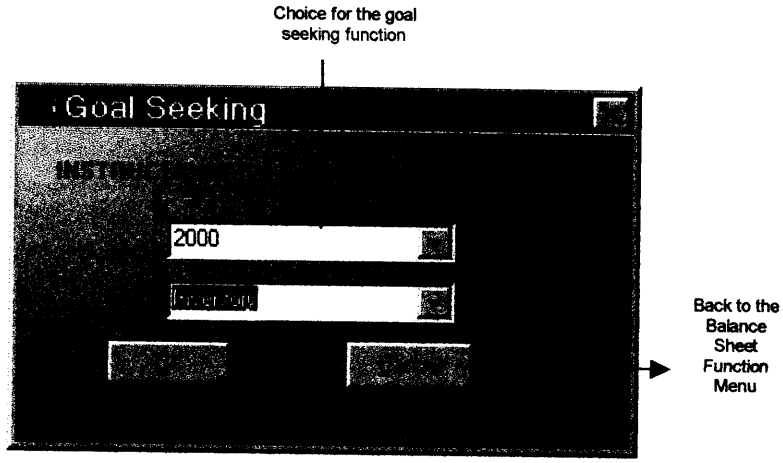


Figure 15: Goal Seeking

4.4.2.1 Output Of Goal Seeking

The input that needs to fill in:

- Target Inventory

This form enables the decision maker to compare the actual liquidity ratios to the target liquidity ratios of the firm. The target liquidity ratios are the result of the target inventory.

Function of the buttons:

To seek for help, user can click ***Help*** button.

The output of the forecasting function:

- Target Liquidity Ratios
- Actual Liquidity Ratios
- Comparison of Target Liquidity Ratios and Actual Liquidity Ratios
- Suggestions for the comparison

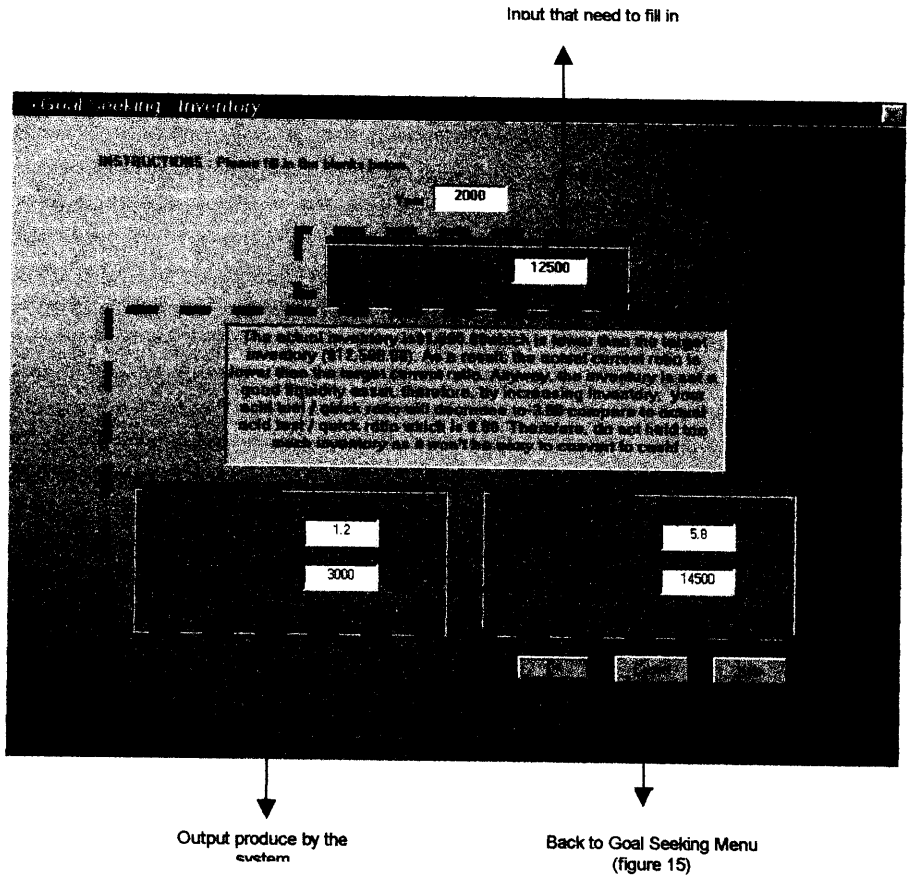
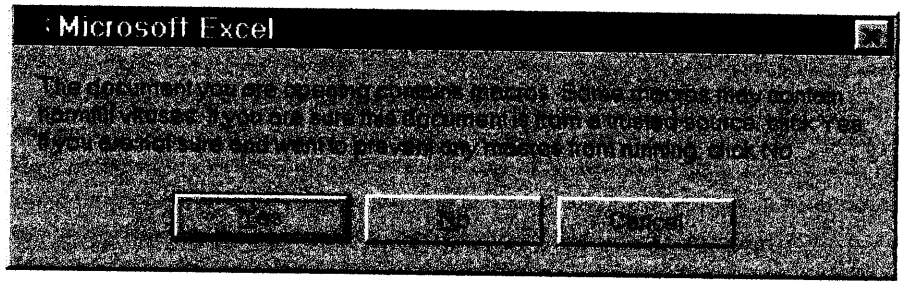


Figure 16: Output Of Goal Seeking

****

User is required to click ***OK*** button when the following form shows.



The login password is "SMEFPDSS".

Appendix B Coding For User Interface In Visual Basics

```
VERSION 5.00
Begin VB.Form

frmMainMenu

Option Explicit
Private Sub cmdOK_Click()
frmLogin.Show
End Sub
**************************************************************************

frmLogin

Option Explicit
Private Sub cmdExit_Click()
Unload Me
Unload frmMainMenu
End Sub

Private Sub cmdOK_Click()
If txtPassword.Text = "SMEFPDSS" Then
Unload Me
Unload frmMainMenu
frmFinancialDecisionSupport.Show
Else
 Beep
 Message$ = "You have entered the wrong password. Please try again!"
 MsgBox Message$, vbExclamation + vbOKOnly, "Wrong Password "
 txtPassword.Text = vbNullString
 txtPassword.SetFocus
End If
End Sub
**************************************************************************

frmFinancialDecisionSupport

Option Explicit
Private Sub cmdExit_Click()
Unload frmFinancialDecisionSupport
End Sub

Private Sub cmdOK_Click()
If (optLoanPaymentAnalysis = False) And (optOperatingBudget = False) And _
  (OptCapitalBudgeting = False) And (optBalanceSheet = False) And _
  (optCashFlow = False) Then
  Message$ = "Oouch..... !"
  Message$ = Message$ & vbCrLf
  Message$ = Message$ & "You haven't choose your choice! "
  Message$ = Message$ & vbCrLf
  MsgBox Message$, vbExclamation + vbOKOnly, " Invalid Choice Entered "
Else
  If (optLoanPaymentAnalysis = False) And (optOperatingBudget = False) And _
```

```
    (optBalanceSheet = False) And (optCashFlow = False) Then
  'Unload frmFinancialDecisionSupport
  frmCapitalBudgeting.Show
  Else
    If (optOperatingBudget = False) And (optBalanceSheet = False) And _
      (optCashFlow = False) Then
    frmLoanPayment.Show
    Else
     If (optBalanceSheet = False) And (optCashFlow = False) Then
     frmOperatingBudgeting1.Show
     Else
      If (optCashFlow = False) Then
      frmBalanceSheet.Show
      Else
      Message$ = "Sorry..... !"
      Message$ = Message$ & vbCrLf
      Message$ = Message$ & "Still Under Construction! "
      Message$ = Message$ & vbCrLf
      MsgBox Message$, vbExclamation + vbOKOnly, "  In Process  "
      End If
     End If
    End If
  End If
End If
End Sub

**************************************************************************

frmBalanceSheet

Option Explicit
 Dim TotalAssets As Currency
 Dim TotalLiablitiesAndEquity As Currency

Private Sub cmdCancel_Click()
   txtYear = vbNullString
   txtCash = vbNullString
   txtAccountReceivable = vbNullString
   txtInventories = vbNullString
   txtNetFixedAssets = vbNullString
   txtAccountPayables = vbNullString
   txtNotesPayables = vbNullString
   txtAccruals = vbNullString
   txtLongTermDebt = vbNullString
   txtOwnerEquity = vbNullString
   txtRetainedEarnings = vbNullString
End Sub

Private Sub AccountPayables_Click()
frmBalanceSheetAccountPayables.Show
End Sub

Private Sub AccountReceivables_Click()
frmBalanceSheetAccountReceivables.Show
End Sub
```

```
Private Sub Accruals_Click()
frmBalanceSheetAccruals.Show
End Sub

Private Sub Cash_Click()
frmBalanceSheetCash.Show
End Sub

Private Sub cmdHelp_Click()
frmBalanceSheetHelp.Show
End Sub

Private Sub cmdMenu_Click()
frmFinancialDecisionSupport.Show
Unload frmBalanceSheet
End Sub

Private Sub cmdOK_Click()
On Error Resume Next
Err.Clear
If Not (IsNumeric(txtYear) And IsNumeric(txtAccountPayables) And_
  IsNumeric(txtAccountReceivable) And IsNumeric(txtAccruals) And _
  IsNumeric(txtCash) And IsNumeric(txtInventories) And _
  IsNumeric(txtLongTermDebt) And IsNumeric(txtNetFixedAssets) And_
  IsNumeric(txtNotesPayables) And IsNumeric(txtOwnerEquity) And_
  IsNumeric(txtRetainedEarnings)) Then


  Message$ = "Oouch...!"
  Message$ = Message$ & vbCrLf
  Message$ = Message$ & " You haven't finish fill in the details."
  MsgBox Message$, vbExclamation + vbOKOnly, " Incompelete Details "
Else

Set obExcelApp = GetObject(, "Excel.Application")
  If Err.Number <> 0 Then
    Err.Clear
    Set obExcelApp = CreateObject("Excel.Application")
    If Err.Number <> 0 Then
      MsgBox "Error: " & Err.Description
    Else
      MyExcel = True
    End If
  Else
    MyExcel = False
  End If

  With obExcelApp
    Set obWorkbooks = GetObject("C:\Project\BalanceSheet.xls")
    With obWorkbooks
      Set obWorkSheet = obWorkbooks.Worksheets("BalanceSheet")
        obWorkSheet.Range("$D$3") = txtYear
        obWorkSheet.Range("$D$5") = txtCash
        obWorkSheet.Range("$D$6") = txtAccountReceivable
```

```
        obWorkSheet.Range("$D$7") = txtInventories
        obWorkSheet.Range("$D$8") = txtNetFixedAssets
        obWorkSheet.Range("$D$11") = txtAccountPayables
        obWorkSheet.Range("$D$12") = txtNotesPayables
        obWorkSheet.Range("$D$13") = txtAccruals
        obWorkSheet.Range("$D$14") = txtLongTermDebt
        obWorkSheet.Range("$D$17") = txtOwnerEquity
        obWorkSheet.Range("$D$18") = txtRetainedEarnings

        TotalAssets = obWorkSheet.Range("$D$28")
        TotalLiablitiesAndEquity = obWorkSheet.Range("$D$38")

  If TotalAssets <> TotalLiablitiesAndEquity Then
     Message$ = "Hmm...!"
     Message$ = Message$ & vbCrLf
     Message$ = Message$ & "The total assets must equals to total liablities and
                equity!"
     MsgBox Message$, vbInformation + vbOKOnly, " Inaccurate Information "
  Else
     frmBalanceSheetRatioAnalysis.Show
  End If
      End With
      obWorkbooks.Save = True
      obWorkbooks.Quit = True
   End With
   obExcelApp.Visible = False
   obExcelApp.Save = True
End If
Unload frmFinancialDecisionSupport
End Sub

Private Sub Definition_Click()
frmBalanceSheetDefinition.Show
End Sub

Private Sub Inventories_Click()
frmBalanceSheetInventories.Show
End Sub

Private Sub LongTermDebt_Click()
frmBalanceSheetLongtermDebt.Show
End Sub

Private Sub NetFixedAssets_Click()
frmBalanceSheetNetFixedAssets.Show
End Sub

Private Sub NotesPayables_Click()
frmBalanceSheetNotesPayables.Show
End Sub

Private Sub OwnerEquity_Click()
frmBalanceSheetOwnerEquity.Show
End Sub
```

```
Private Sub RetainedEarnings_Click()
frmBalanceSheetRetainedEarning.Show
End Sub

Private Sub txtAccountPayables_Change()
If ((txtAccountPayables.Text) <> " ") Then
 If IsNumeric(txtAccountPayables.Text) Then
  'do nothing
 Else
  Beep
  Message$ = "Oouch...!"
  Message$ = Message$ & vbCrLf
  Message$ = Message & "The account payables element" & _
        "must numeric value (e.g. RM 2000)."
  MsgBox Message, vbCritical + vbOKOnly, " Invalid Account Payables Values
        Entry"
  txtAccountPayables.SetFocus
 End If
Else
'do nothing
End If
End Sub

Private Sub txtAccountReceivable_Change()
If ((txtAccountReceivable.Text) <> " ") Then
 If IsNumeric(txtAccountReceivable.Text) Then
  'do nothing
 Else
  Beep
  Message$ = "Oouch...!"
  Message$ = Message$ & vbCrLf
  Message$ = Message & "The account receivable element" & _
        "must numeric value (e.g. RM 2000)."
  MsgBox Message, vbCritical + vbOKOnly, " Invalid Account Receivable Values
Entry "
  txtAccountReceivable.SetFocus
 End If
Else
'do nothing
End If
End Sub

Private Sub txtAccruals_Change()
If ((txtAccruals.Text) <> " ") Then
 If IsNumeric(txtAccruals.Text) Then
  'do nothing
 Else
  Beep
  Message$ = "Oouch...!"
  Message$ = Message$ & vbCrLf
  Message$ = Message & "The accruals element must in numeric value (e.g. RM
        2000)."
  MsgBox Message, vbCritical + vbOKOnly, " Invalid Accruals Values Entry "
  txtAccruals.SetFocus
 End If
```

```
Else
 'do nothing
End If
End Sub

Private Sub txtCash_Change()
If ((txtCash.Text) <> " ") Then
  If IsNumeric(txtCash.Text) Then
    'do nothing
  Else
    Beep
    Message$ = "Oouch...!"
    Message$ = Message$ & vbCrLf
    Message$ = Message & "The cash element must in numeric value (e.g. RM 2000)."
    MsgBox Message, vbCritical + vbOKOnly, " Invalid Cash Values Entry "
    txtCash.SetFocus
  End If
Else
 'do nothing
End If
End Sub
Private Sub txtInventories_Change()
If ((txtInventories.Text) <> " ") Then
  If IsNumeric(txtInventories.Text) Then
    'do nothing
  Else
    Beep
    Message$ = "Oouch...!"
    Message$ = Message$ & vbCrLf
    Message$ = Message & "The inventories element must in numeric value (e.g. RM
               2000)."
    MsgBox Message, vbCritical + vbOKOnly, " Invalid Inventories Values Entry "
    txtInventories.SetFocus
  End If
Else
 'do nothing
End If
End Sub

Private Sub txtLongTermDebt_Change()
If ((txtLongTermDebt.Text) <> " ") Then
  If IsNumeric(txtLongTermDebt.Text) Then
    'do nothing
  Else
    Beep
    Message$ = "Oouch...!"
    Message$ = Message$ & vbCrLf
    Message$ = Message & "The long term debt element must in numeric value (e.g. RM
               &_2000)."
    MsgBox Message, vbCritical + vbOKOnly, " Invalid Long Term Debt Values Entry "
    txtLongTermDebt.SetFocus
  End If
Else
 'do nothing
End If
```

```
End Sub

Private Sub txtNetFixedAssets_Change()
If ((txtNetFixedAssets.Text) <> " ") Then
 If IsNumeric(txtNetFixedAssets.Text) Then
  'do nothing
 Else
  Beep
  Message$ = "Oouch...!"
  Message$ = Message$ & vbCrLf
  Message$ = Message & "The net fixed assets element must in numeric value (e.g.
             RM &_ 2000)."
  MsgBox Message, vbCritical + vbOKOnly, " Invalid Net Fixed Assets Values Entry "
  txtNetFixedAssets.SetFocus
 End If
Else
'do nothing
End If
End Sub

Private Sub txtNotesPayables_Change()
If ((txtNotesPayables.Text) <> " ") Then
 If IsNumeric(txtNotesPayables.Text) Then
  'do nothing
 Else
  Beep
  Message$ = "Oouch...!"
  Message$ = Message$ & vbCrLf
  Message$ = Message & "The notes payables element must in numeric value (e.g. RM
             &_ 2000)."
  MsgBox Message, vbCritical + vbOKOnly, " Invalid Notes Payables Entry "
  txtNotesPayables.SetFocus
 End If
Else
'do nothing
End If
End Sub

Private Sub txtOwnerEquity_Change()
If ((txtOwnerEquity.Text) <> " ") Then
 If IsNumeric(txtOwnerEquity.Text) Then
  'do nothing
 Else
  Beep
  Message$ = "Oouch...!"
  Message$ = Message$ & vbCrLf
  Message$ = Message & "The owner equity element must in numeric value (e.g. RM
2000)."
  MsgBox Message, vbCritical + vbOKOnly, " Invalid Owner Equity Entry "
  txtOwnerEquity.SetFocus
 End If
Else
'do nothing
End If
End Sub
```

```
Private Sub txtRetainedEarnings_Change()
If ((txtRetainedEarnings.Text) <> " ") Then
 If IsNumeric(txtRetainedEarnings.Text) Then
  'do nothing
 Else
  Beep
  Message$ = "Oouch...!"
  Message$ = Message$ & vbCrLf
  Message$ = Message & "The retained earnings element must in numeric value (e.g.
RM &_
            2000)."
  MsgBox Message, vbCritical + vbOKOnly, " Invalid Retained Earnings Entry "
  txtRetainedEarnings.SetFocus
 End If
Else
'do nothing
End If

End Sub
Private Sub txtYear_Change()
If ((txtYear.Text) <> " ") Then
 If IsNumeric(txtYear.Text) Then
  'do nothing
 Else
  Beep
  Message$ = "Oouch...!"
  Message$ = Message$ & vbCrLf
  Message$ = Message & "The year element must in numeric value (e.g. RM 2000)."
  MsgBox Message, vbCritical + vbOKOnly, " Invalid txtYear Entry "
  txtYear.SetFocus
 End If
Else
'do nothing
End If
End Sub

Private Sub txtYear_LostFocus()
If IsNumeric(txtYear.Text) Then
 If ((txtYear.Text) >= 1990) And ((txtYear.Text) <= 2000) Then
  'do nothing
 Else
  Beep
  Message$ = "Hmmm...!"
  Message$ = Message$ & vbCrLf
  Message$ = Message & "The year element must in range between 1990 and 2000."
  MsgBox Message, vbExclamation + vbOKOnly, " Invalid txtYear Entry "
  txtYear.SetFocus
  End If
End If
End Sub

**************************************************************************

frmBalanceSheetRatioAnalysis
```

```
Option Explicit
Private Sub cmdCancel_Click()
Unload Me
End Sub

Private Sub cmdOK_Click()
If (optForecasting = False) And (optGoalSeeking = False) Then
  Message$ = "Oouch..... !"
  Message$ = Message$ & vbCrLf
  Message$ = Message$ & "You haven't choose your choice! "
  Message$ = Message$ & vbCrLf
  MsgBox Message$, vbExclamation + vbOKOnly, " Invalid Choice Entered "
Else
   If (optGoalSeeking = False) Then
   Unload Me
   frmBalanceSheetForecasting.Show
   Else
     optForecasting = False
     Unload Me
     frmBalanceSheetGoalSeeking.Show
   End If
End If
End Sub

**************************************************************************

frmBalanceSheetForecasting

Option Explicit
Private Sub cmdCancel_Click()
Unload Me
End Sub

Private Sub cmdOK_Click()
On Error Resume Next
Err.Clear
   Set obExcelApp = GetObject(, "Excel.Application")
   If Err.Number <> 0 Then
   Err.Clear
      Set obExcelApp = CreateObject("Excel.Application")
      If Err.Number <> 0 Then
        MsgBox "Error: " & Err.Description
      Else
        MyExcel = True
      End If
   Else
      MyExcel = False
   End If

If (Combo1.ListIndex = -1) Then
  Message$ = "Oouch...!"
  Message$ = Message$ & vbCrLf
  Message$ = Message$ & " You haven't enter your choice!"
  MsgBox Message$, vbCritical + vbOKOnly, " Invalid Choice Entered "
```

```
Else
 If (Combo1.ListIndex = 0) Then
  If Not (IsNumeric(txtInflationRate)) Then
   Message$ = "Oouch...!"
   Message$ = Message$ & vbCrLf
   Message$ = Message$ & " Invalid Inlfation Rate!"
   MsgBox Message$, vbInformation + vbOKOnly, " Invalid Information"
  Else
   With obExcelApp
    Set obWorkbooks = GetObject("C:\Project\BalanceSheet.xls")
    With obWorkbooks
     Set obWorkSheet = obWorkbooks.Worksheets("BalanceSheet")
         obWorkSheet.Range("$I$4") = txtInflationRate
    End With
     obWorkbooks.Save = True
     obWorkbooks.Close = True
     obWorkbooks.Quit = True
   End With
    obExcelApp.Visible = False
    obExcelApp.Save = True
    obExcelApp.Close = True
   Unload Me
   frmBalanceSheetLiquidityRatio.Show
  End If
 Else
  Message$ = "Sorry...!"
  Message$ = Message$ & vbCrLf
  Message$ = Message$ & " Still Under Constructions!"
  MsgBox Message$, vbInformation + vbOKOnly, " In Process "
 End If
End If
End Sub

Private Sub Form_Load()
   Combo1.List(0) = "Liquidity"
   Combo1.List(1) = "Debt Management"
   Combo1.List(2) = "Profitability"
End Sub

Private Sub txtInflationRate_LostFocus()
If ((txtInflationRate.Text) <> "") Then
 If (IsNumeric(txtInflationRate.Text)) Then
    txtInflationRate.Text = ((txtInflationRate.Text) / 100#)
   If (((txtInflationRate) >= 0) And ((txtInflationRate) <= 100)) Then
    'do nothing
   Else
    Beep
    Message$ = "Hmmm...!"
    Message$ = Message$ & vbCrLf
    Message$ = Message$ & "I'm sorry that your request is beyond my forecasting
ablity! "
    MsgBox Message$, vbCritical + vbOKOnly, " Wrong Inflation Rate Range "
    txtInflationRate.SetFocus
   End If
  Else
```

```
    Beep
    Message$ = "Oouch...!"
    Message$ = Message$ & vbCrLf
    Message$ = Message & "The inflation rate must in numeric value (e.g. 12)."
    MsgBox Message, vbCritical + vbOKOnly, " Wrong Inflation Rate Range Entry "
    txtInflationRate.SetFocus
  End If
Else
  'do nothing
End If
End Sub
**************************************************************************

frmBalanceSheetGoalSeeking

Option Explicit
 Dim IndustryCurrentRatio As Currency
 Dim IndustryAcidTest As Currency
 Dim CurrentRatio As Currency
 Dim AcidTest As Currency
 Dim DifferentAcidTest As Currency
 Dim DifferentCurrentRatio As Currency

Private Sub cmdCancel_Click()
Unload Me
frmBalanceSheetForecasting.Show
End Sub

Private Sub cmdHelp_Click()
frmBalanceSheetLiquidityRatioHelp.Show
End Sub

Private Sub cmdOK_Click()
On Error Resume Next
Err.Clear
Set obExcelApp = GetObject(, "Excel.Application")
   If Err.Number <> 0 Then
      Err.Clear
      Set obExcelApp = CreateObject("Excel.Application")
      If Err.Number <> 0 Then
         MsgBox "Error: " & Err.Description
      Else
         MyExcel = True
      End If
   Else
      MyExcel = False
   End If
   With obExcelApp
      Set obWorkbooks = GetObject("C:\Project\BalanceSheet.xls")
      With obWorkbooks
        Set obWorkSheet = obWorkbooks.Worksheets("BalanceSheet")
          CurrentRatio = obWorkSheet.Range("$D$43")
          AcidTest = obWorkSheet.Range("$D$44")
```

```
IndustryCurrentRatio = CCur(txtIndustryCurrentRatio)
IndustryAcidTest = CCur(txtIndustryAcidTest)

DifferentAcidTest = IndustryAcidTest - AcidTest
DifferentCurrentRatio = IndustryCurrentRatio - CurrentRatio

If Not (IsNumeric(txtIndustryCurrentRatio) And IsNumeric(txtIndustryAcidTest))
Then
  Message$ = "Oouch....!"
  Message$ = Message$ & vbCrLf
  Message$ = Message$ & " You haven't entered the industry average ratio !"
  MsgBox Message$, vbInformation + vbOKOnly, " Inaccurate Information"

  Else
    Frame1.Visible = True
    If (AcidTest < IndustryAcidTest And CurrentRatio < IndustryCurrentRatio)
    Then
      txtAnalysisStatement = "Your company liquidity ratio is poor compare to
      the industry average ratio. Beware of it! You may face problem if you don't
      have enough cash to pay short term debt! It may trouble to covert your asset
      to cash" & _" without having to reduce the price of the asset!"
      txtAnalysisStatement.BackColor = &HC0C0FF
      If (DifferentAcidTest > 1) Then
        txtSuggestion = "You hold too much inventories! Reduce your inventory
       in hand! It help you save cash in holding inventory cost."
        txtSuggestion.BackColor = &HFFFF80
      Else
        If (DifferentCurrentRatio > 1) Then
          txtSuggestion = "Reduce your asset in hand!" & _
          " Too much asset in hand may cause you hard to convert cash " & _
           "when you are in emergency!"
          txtSuggestion.BackColor = &HFFFF80
        Else
          'do nothing
        End If
      End If
     Else
     If (AcidTest > IndustryAcidTest And CurrentRatio < IndustryCurrentRatio)
    Then txtAnalysisStatement = " Beware! Your company acid test / quick ratio is
    better compare to the industry average ratio. Anyway! Your company current
    ratio is  poor  then industry average! "
          txtAnalysisStatement.BackColor = &HC0FFFF
          txtSuggestion = "Keep an eye to current ratio." & _
          " You may reduce your current asset in hand. It helps your company to
          get cash in order to increase the current ratio!"
           txtSuggestion.BackColor = &HFFFF80

        Else
        If (AcidTest < IndustryAcidTest And CurrentRatio >
        IndustryCurrentRatio) _
          Then
            txtAnalysisStatement = " Beware! Your company current ratio is better
            compare to the industry average ratio. Anyway! Your company acid
            test/ quick ratio is poor then industry average! "
             txtAnalysisStatement.BackColor = &HC0FFFF
```

```
                txtSuggestion = "Keep an eye to the acid test / quick ratio." & _
                " You may reduce your inventories in hand. It helps your company to
                get cash
                in order to increase the acid test / quick ratio!"
                txtSuggestion.BackColor = &HFFFF80
              Else
                If (AcidTest > IndustryAcidTest And CurrentRatio >
                IndustryCurrentRatio)_ Then
                  txtAnalysisStatement = "Congratulations! Your company liquidity
                  ratio is better compare to the industry average ratio. Maintain the
                  liquidity ratio " &_ " of your company! Anyway, do not keep too
                  much cash in hand " & _
                    " Cash is considered as non productive asset!"
                    txtAnalysisStatement.BackColor = &HC0FFC0
                    txtSuggestion = "Keep an eye to your liquidity ratio." & _
                    " Make sure the ratio is around the industry average ratio. " & _
                    " Too high or too low may cause problem to your company!"
                    txtSuggestion.BackColor = &HFFFF80
                End If
              End If
            End If
          End If
        End If
      End With
        obWorkbooks.Save = True
        obWorkbooks.Quit = True
    End With
     obExcelApp.Visible = False
     obExcelApp.Save = True
End Sub

Private Sub Form_Load()
Frame1.Visible = False
On Error Resume Next
Err.Clear
Set obExcelApp = GetObject(, "Excel.Application")
  If Err.Number <> 0 Then
    Err.Clear
     Set obExcelApp = CreateObject("Excel.Application")
     If Err.Number <> 0 Then
       MsgBox "Error: " & Err.Description
     Else
       MyExcel = True
     End If
  Else
    MyExcel = False
  End If
  With obExcelApp
    Set obWorkbooks = GetObject("C:\Project\BalanceSheet.xls")
    With obWorkbooks
     Set obWorkSheet = obWorkbooks.Worksheets("BalanceSheet")
        CurrentRatio = obWorkSheet.Range("$D$43")
        AcidTest = obWorkSheet.Range("$D$44")
        txtCurrentRatio = Format(CurrentRatio, "0.00")
        txtAcidTest = Format(AcidTest, "0.00")
```

```
      End With
      obWorkbooks.Save = True
      obWorkbooks.Quit = True
    End With
    obExcelApp.Visible = False
    obExcelApp.Save = True
End Sub
**************************************************************************

frmBalanceSheetGoalSeeking

Option Explicit
 Dim Year0 As Integer
 Dim Year1 As Integer
 Dim Year2 As Integer
 Dim Year3 As Integer
 Dim Year4 As Integer
 Dim Year5 As Integer

Private Sub cmdCancel_Click()
Unload Me
End Sub

Private Sub cmdOK_Click()
If (Combo1.ListIndex = -1 Or Combo2.ListIndex = -1) Then
  Message$ = "Oouch...!"
  Message$ = Message$ & vbCrLf
  Message$ = Message$ & " You haven't enter the choice! "
  MsgBox Message$, vbInformation + vbOKOnly, " Invalid Choice Enter"
Else
  If (Combo1.Text = Year0) And (Combo2.Text = "Inventory") Then
  frmBalanceSheetInventoryYear0.Show
  Else
    Message$ = "Sorry...!"
    Message$ = Message$ & vbCrLf
    Message$ = Message$ & " Still Under Construction !"
    MsgBox Message$, vbInformation + vbOKOnly, " In Process"
  End If
End If
End Sub

Private Sub Form_Load()
On Error Resume Next
Err.Clear
    Set obExcelApp = GetObject(, "Excel.Application")
    If Err.Number <> 0 Then
    Err.Clear
       Set obExcelApp = CreateObject("Excel.Application")
       If Err.Number <> 0 Then
         MsgBox "Error: " & Err.Description
       Else
         MyExcel = True
       End If
    Else
       MyExcel = False
```

```
  End If
  With obExcelApp
    Set obWorkbooks = GetObject("C:\Project\BalanceSheet.xls")
    With obWorkbooks
      Set obWorkSheet = obWorkbooks.Worksheets("BalanceSheet")
        Year0 = obWorkSheet.Range("$D$21")
        Year1 = obWorkSheet.Range("$E$21")
        Year2 = obWorkSheet.Range("$F$21")
        Year3 = obWorkSheet.Range("$G$21")
        Year4 = obWorkSheet.Range("$H$21")
        Year5 = obWorkSheet.Range("$I$21")

        Combo1.AddItem Year0
        Combo1.AddItem Year1
        Combo1.AddItem Year2
        Combo1.AddItem Year3
        Combo1.AddItem Year4
        Combo1.AddItem Year5

        Combo2.List(0) = "Inventory"
        Combo2.List(1) = "Long-term debt"
        Combo2.List(2) = "Retained Earning"
      End With
      obWorkbooks.Close = True
      obWorkbooks.Save = True
      obWorkbooks.Quit = True
    End With
    obExcelApp.Close = True
    obExcelApp.Save = True
End Sub

**************************************************************************

frmBalanceSheetInventoryYear0

Option Explicit
 Dim CurrentAsset As Currency
 Dim CurrentRatio As Currency
 Dim CurrentAssetGoal As Currency
 Dim CurrentRatioGoal As Currency
 Dim ActualInventory As Currency
 Dim AcidTest As Currency
 Dim ActualAcidTest As Currency

Private Sub cmdOK_Click()
On Error Resume Next
Err.Clear
    Set obExcelApp = GetObject(, "Excel.Application")
    If Err.Number <> 0 Then
    Err.Clear
      Set obExcelApp = CreateObject("Excel.Application")
      If Err.Number <> 0 Then
        MsgBox "Error: " & Err.Description
      Else
        MyExcel = True
```

```
    End If
  Else
    MyExcel = False
  End If

If (IsNumeric(txtInventoryGoal.Text)) Then
 If ((txtInventoryGoal.Text) > 0) Then
  With obExcelApp
   Set obWorkbooks = GetObject("C:\Project\BalanceSheet.xls")
   With obWorkbooks
    Set obWorkSheet = obWorkbooks.Worksheets("BalanceSheet")
      obWorkSheet.Range("$D$61") = txtInventoryGoal

       CurrentAsset = obWorkSheet.Range("$D$26")
       CurrentRatio = obWorkSheet.Range("$D$43")
       CurrentAssetGoal = obWorkSheet.Range("$D$63")
       CurrentRatioGoal = obWorkSheet.Range("$D$62")
       ActualInventory = obWorkSheet.Range("$D$25")
       AcidTest = obWorkSheet.Range("$D$64")
       ActualAcidTest = obWorkSheet.Range("$D$44")

       txtCurrentAsset = CurrentAsset
       txtCurrentRatio = CurrentRatio
       txtCurrentAssetGoal = CurrentAssetGoal
       txtCurrentRatioGoal = CurrentRatioGoal

    If (CurrentAsset < CurrentAssetGoal) And (CurrentRatio < CurrentRatioGoal)
Then
      txtAnalysis = "The actual inventory is" & Format(ActualInventory, "Currency")
& _
      "which is lower than the target inventory (" & Format(txtInventoryGoal,
      "Currency). As a result, the actual current ratio is lower " & _"than the target
current
      ratio. Anyway, the inventory is not a good liquidity asset, therefore, by
increasing
      inventory, " & _" your acid test / quick ratio will decrease to" &
Format(AcidTest,
      "0.00") & " compare to actual acid test / quick ratio" & _" which is " &
      Format(ActualAcidTest, "0.00") & ". Therefore, do not held too much inventory
as it
      won't be easy to convert to cash!"
      txtAnalysis.BackColor = &HC0C0FF
    Else
      If (CurrentAsset > CurrentAssetGoal) And (CurrentRatio > CurrentRatioGoal)
Then
       txtAnalysis = "The actual inventory is" & Format(ActualInventory,
"Currency") & _
       "which is higher than the target inventory (" & Format(txtInventoryGoal,
       "Currency. As a result, the current ratio is higher than the target current ratio"
& _
       ". However, the inventory is not a good liquidity asset. Athough the current
ratio has
       decreased, " & _" the acid test / quick ratio has increased to" &
Format(AcidTest,
       "0.00") & ". compare to the actual acid test /" & _" quick ratio which is " &
```

```
            Format(ActualAcidTest, "0.00") & ". "
            txtAnalysis.BackColor = &HFFFFC0
          Else
            'do nothing
          End If
        End If
        End With
        obWorkbooks.Quit
        obWorkbooks.Save = True
        obWorkbooks.Close = True
      End With
      obExcelApp.Save = True
      obExcelApp.Close = True
      obExcelApp.Visible = False
      txtAnalysis.Visible = True
      Frame1.Visible = True
      Frame2.Visible = True
    Else
      Message$ = "Oouch...!"
      Message$ = Message$ & vbCrLf
      Message$ = Message$ & " The inventory goal must in positive value ! (e.g RM
20,000)"
      MsgBox Message$, vbInformation + vbOKOnly, " Invalid Goal Inventory Entry"
    End If
  Else
    Message$ = "Oouch...!"
    Message$ = Message$ & vbCrLf
    Message$ = Message$ & " The inventory goal must in the numeric value ! (e.g RM
20,000)"
    MsgBox Message$, vbInformation + vbOKOnly, " Invalid Goal Inventory Entry"
  End If
  End Sub

  Private Sub cmdCancel_Click()
  Unload Me
  End Sub

  Private Sub cmdHelp_Click()
  frmBalanceSheetGoalSeekingInventoryYear0Help.Show
  End Sub

  Private Sub Form_Load()
  On Error Resume Next
  Err.Clear
    Set obExcelApp = GetObject(, "Excel.Application")
    If Err.Number <> 0 Then
    Err.Clear
      Set obExcelApp = CreateObject("Excel.Application")
      If Err.Number <> 0 Then
        MsgBox "Error: " & Err.Description
      Else
        MyExcel = True
      End If
    Else
      MyExcel = False
```

```
  End If
  With obExcelApp
    Set obWorkbooks = GetObject("C:\Project\BalanceSheet.xls")
    With obWorkbooks
      Set obWorkSheet = obWorkbooks.Worksheets("BalanceSheet")
        txtYear = obWorkSheet.Range("$D$21")
    End With
    obWorkbooks.Save = True
    obWorkbooks.Close = True
  End With
  obExcelApp.Visible = False
  obExcelApp.Close = True
  obExcelApp.Save = True
txtAnalysis.Visible = False
Frame1.Visible = False
Frame2.Visible = False
End Sub

Private Sub txtInventoryGoal_Change()
If ((txtInventoryGoal.Text) <> "") Then
 If (IsNumeric(txtInventoryGoal.Text)) Then
   If ((txtInventoryGoal.Text) > 0) Then
    'do nothing
   Else
    Beep
    Message$ = "Oouch...!"
    Message$ = Message$ & vbCrLf
    Message$ = Message & "The asset must have value (e.g. RM 15,000)."
    MsgBox Message, vbCritical + vbOKOnly, " Wrong Inventory Goal Value Entry "
    txtInventoryGoal.SetFocus
   End If
 Else
  Beep
  Message$ = "Oouch...!"
  Message$ = Message$ & vbCrLf
  Message$ = Message & "The asset must have in numeric value (e.g. RM 15,000)."
  MsgBox Message, vbCritical + vbOKOnly, " Wrong Inventory Goal Value Entry "
  txtInventoryGoal.SetFocus
 End If
Else
 'do nothing
End If
End Sub
**************************************************************************

frmCapitalBudgeting

Option Explicit
Private Sub AssetYearLast_Click()
frmCapitalBudgetingAssetYearLast.Show
End Sub

Private Sub cmdAnalysis_Click()
If IsNumeric(txtAssetYearBuy) And IsNumeric(txtDuration) And IsNumeric(txtIRR)
And IsNumeric(txtAssetCost) Then
```

```
  AssetYearBuy = CInt(txtAssetYearBuy)
  Duration = CInt(txtDuration)
  IRR = CInt(txtIRR)
  AssetCost = CCur(txtAssetCost)
  frmCapitalBudgetingAnalysis.Show
Else
  Message$ = "Oouch...!"
  Message$ = Message$ & vbCrLf
  Message$ = Message$ & "You haven't complete the information needed!"
  MsgBox Message$, vbCritical + vbOKOnly, " Incomplete Information "
End If
End Sub

Private Sub cmdHelp_Click()
frmCapitalBudgetingHelp.Show
End Sub

Private Sub cmdMenu_Click()
Unload frmCapitalBudgeting
frmFinancialDecisionSupport.Show
End Sub

Private Sub cmdOK_Click()
On Error Resume Next
Err.Clear
   Set obExcelApp = GetObject(, "Excel.Application")
   If Err.Number <> 0 Then
   Err.Clear
      Set obExcelApp = CreateObject("Excel.Application")
      If Err.Number <> 0 Then
        MsgBox "Error: " & Err.Description
      Else
       MyExcel = True
      End If
   Else
     MyExcel = False
   End If

If IsNumeric(txtAssetYearBuy) And IsNumeric(txtDuration) And IsNumeric(txtIRR)
And _
  IsNumeric(txtAssetCost) And IsNumeric(txtCurrentInterestRate) Then

With obExcelApp
   Set obWorkbooks = GetObject("C:\Project\CapitalBudgeting2.xls")
    With obWorkbooks
    Set obWorkSheet = obWorkbooks.Worksheets("CapitalBudgeting")
    obWorkSheet.Range("$D$4").Value = txtAssetCost
    obWorkSheet.Range("$D$5").Value = txtAssetYearBuy
    obWorkSheet.Range("$D$6").Value = txtDuration
    obWorkSheet.Range("$D$7").Value = txtIRR
    obWorkSheet.Range("$D$8").Value = txtCurrentInterestRate

    txtCapitalPayback = Format((obWorkSheet.Range("$I$4").Value), "0.00")
    txtAssetYearLast = obWorkSheet.Range("$I$7").Value
    txtIndexProfit = Format((obWorkSheet.Range("$I$5").Value), "0.00")
```

```
    txtNPV = Format((obWorkSheet.Range("$I$6")), "0.00")
    frameOutput.Visible = True

    Set ChartData = obWorkbooks.Worksheets("ChartData")
      ChartData.Range("B2:B22").NumberFormat = "0.00"
    Set ChartSheet = obWorkbooks.Charts.Add
      ChartSheet.ChartType = xlColumnClustered
      ChartSheet.SetSourceData ChartData.Range("B2:B22"), xlColumns
      ChartSheet.SeriesCollection(1).XValues = "=ChartData!R2C1:R22C1"
      ChartSheet.SeriesCollection(1).Values = "=ChartData!R2C2:R22C2"
      ChartSheet.SeriesCollection(1).Name = "=ChartData!R1C2"
      ChartSheet.HasTitle = True
      ChartSheet.ChartTitle.Characters.Text = "Net Present Value For Capital Bedgeting"
      ChartSheet.Axes(xlCategory, xlPrimary).HasTitle = True
      ChartSheet.Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = "Year"
      ChartSheet.Axes(xlValue, xlPrimary).HasTitle = True
      ChartSheet.Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "NPV"
      ChartSheet.ChartArea.Copy
      Image1.Picture = Clipboard.GetData(vbCFBitmap)
  End With
  obWorkbooks.Save = True
  obWorkbooks.Close = True
  obWorkbooks.Quit = True
End With
  obExcelApp.Save = True
  obExcelApp.Close = True
Else
  Beep
  Message$ = "Oouch...!"
  Message$ = Message$ & vbCrLf
  Message$ = Message$ & "You haven't complete the data!"
  MsgBox Message$, vbCritical + vbOKOnly, "Incomplete Data Entry"
End If
Unload frmFinancialDecisionSupport
End Sub

Private Sub CostOfAsset_Click()
frmCapitalBudgetingCostOfAsset.Show
End Sub
Private Sub CurrentInterestRate_Click()
frmCapitalBudegtingCurrentInterest.Show
End Sub

Private Sub Definition_Click()
frmCapitalBudgetingDefinition.Show
End Sub

Private Sub Duration_Click()
frmCapitalBudgetingDuration.Show
End Sub

Private Sub Form_Load()
frameOutput.Visible = False
End Sub
```

```
Private Sub IndexProfitability_Click()
frmCapitalBudgetingIndexProfitability.Show
End Sub

Private Sub IRR_Click()
frmCapitalBudgetingMinimumRequiredInterestRate.Show
End Sub

Private Sub NPV_Click()
frmCapitalBudgetingNPV.Show
End Sub

Private Sub PaybackPeriod_Click()
frmCapitalBudgetingPaybackPeriod.Show
End Sub

Private Sub txtAssetCost_Change()
If ((txtAssetCost.Text) <> "") Then
 If IsNumeric(txtAssetCost.Text) Then
  'do nothing
 Else
  Beep
  Message$ = "Oouch...!"
  Message$ = Message$ & vbCrLf
  Message$ = Message$ & "The asset cost must be minima in numeric values (e.g.
RM3,000)."
  MsgBox Message$, vbCritical + vbOKOnly, "Wrong Asset Cost Entry"
  txtAssetCost.SetFocus
 End If
Else
 'do nothing
End If
End Sub
Private Sub txtAssetCost_LostFocus()
If (IsNumeric(txtAssetCost.Text)) Then
 If ((txtAssetCost.Text) > 0) Then
  'do nothing
 Else
  Beep
  Message$ = "Oouch...!"
  Message$ = Message$ & vbCrLf
  Message$ = Message & "The asset must have value (e.g. RM 15,000)."
  MsgBox Message, vbCritical + vbOKOnly, " Wrong Years Range Entry "
  txtAssetCost.SetFocus
 End If
Else
 'do nothing
End If
End Sub

Private Sub txtAssetYearBuy_Change()
If ((txtAssetYearBuy.Text) <> "") Then
 If IsNumeric(txtAssetYearBuy.Text) Then
 Else
  Beep
```

```
    Message$ = "Oouch...!"
    Message$ = Message$ & vbCrLf
    Message$ = Message & "The year must be in 4 numeric values (e.g. 1999)."
    MsgBox Message, vbCritical + vbOKOnly, " Wrong Years Values Entry "
    txtAssetYearBuy.SetFocus
  End If
Else
'do nothing
End If
End Sub

Private Sub txtAssetYearBuy_LostFocus()
If (IsNumeric(txtAssetYearBuy.Text)) Then
  If ((txtAssetYearBuy.Text) >= 1990) Then
    'do nothing
  Else
    Beep
    Message$ = "Oouch...!"
    Message$ = Message$ & vbCrLf
    Message$ = Message & "The year must be in range of 1990 to 2000 (e.g. 1999)."
    MsgBox Message, vbCritical + vbOKOnly, " Wrong Years Range Entry "
    txtAssetYearBuy.SetFocus
  End If
Else
  'do nothing
End If
End Sub
Private Sub txtDuration_Change()
If ((txtDuration.Text) <> "") Then
  If IsNumeric(txtDuration.Text) Then
    'do nothing
  Else
    Beep
    Message = "Oouch...!"
    Message = Message & vbCrLf
    Message = Message & "The duration year must be numeric values (e.g. 20)."
    MsgBox Message, vbCritical + vbOKOnly, " Wrong Years Values Entry "
    txtDuration.SetFocus
  End If
Else
  'do nothing
End If
End Sub

Private Sub txtCurrentInterestRate_Change()
If IsNumeric(txtCurrentInterestRate.Text) Then
  'do nothing
Else
  Beep
  Message = "Oouch...!"
  Message = Message1$ & vbCrLf
  Message = Message1$ & "The interest rate must be in numeric values (e.g. 11)."
  MsgBox Message, vbCritical + vbOKOnly, " Wrong Interest Rate Value "
  txtCurrentInterestRate.SetFocus
End If
```

```
End Sub

Private Sub txtIRR_Change()
If IsNumeric(txtIRR.Text) Then
 'do nothing
Else
 Beep
 Message = "Oouch...!"
 Message = Message1$ & vbCrLf
 Message = Message1$ & "The required interest rate must be in numeric values (e.g.
11)."
 MsgBox Message, vbCritical + vbOKOnly, " Wrong Required Interest Rate Value "
 txtIRR.SetFocus
End If
End Sub

Private Sub txtIRR_LostFocus()
If ((txtIRR.Text) <> "") Then
 If (((txtIRR.Text) >= 0) And ((txtIRR.Text) <= 100)) Then
  'do nothing
 Else
  Beep
  Message$ = "Hmmm...!"
  Message$ = Message$ & vbCrLf
  Message$ = Message$ & "I'm sorry that your request is beyond my forecasting
             ability! "
  MsgBox Message$, vbCritical + vbOKOnly, " Wrong Required Interest Rate Range "
  txtIRR.SetFocus
 End If
Else
 'do nothing
End If
End Sub
Private Sub txtCurrentInterestRate_LostFocus()
If ((txtCurrentInterestRate.Text) <> "") Then
 If (((txtCurrentInterestRate.Text) >= 0) And ((txtCurrentInterestRate.Text) <= 100))
Then
  'do nothing
 Else
  Beep
  Message1$ = "Hmmm...!"
  Message1$ = Message1$ & vbCrLf
  Message1$ = Message1$ & "I'm sorry that your request is beyond my forecasting
ablity! "
  MsgBox Message1$, vbCritical + vbOKOnly, " Wrong Interest Rate Range "
  txtCurrentInterestRate.SetFocus
 End If
Else
 'do nothing
End If
End Sub

Private Sub txtDuration_LostFocus()
If ((txtDuration.Text) <> "") Then
 If (((txtDuration.Text) >= 0) And ((txtDuration.Text) <= 20)) Then
```

```
  'do nothing
 Else
  Beep
  Message1$ = "Hmmm...!"
  Message1$ = Message1$ & vbCrLf
  Message1$ = Message1$ & "I'm sorry that your request is beyond my forecasting
ablity! "
  MsgBox Message1$, vbCritical + vbOKOnly, " Wrong Duration Range "
  txtDuration.SetFocus
 End If
Else
 'do nothing
End If
End Sub
**************************************************************************

frmLoanPayment

Option Explicit
Dim FutureValue As Double ' fv
Dim NumberPayment As Double ' nper
Dim PresentValue As Double ' pv
Dim Payment As Double ' Pmt
Dim Rate As Double ' Rate
Dim YearPayment As Double ' years of payment made
Dim Discount As Double 'Compound Discount Rate
Dim NewLoanInterest As Single 'Discount Loan Interest
Dim OldLoanInterest As Single 'Discount Loan Interest
Dim LoanInterest As Single ' Loan Interest
Dim i As Integer

Private Sub AnnnualCurrentInterest_Click()
frmLoanPaymentAnnualCurrentInterest.Show
End Sub

Private Sub AnnualLoanInterest_Click()
frmLoanPaymentAnnualLoanInterest.Show
End Sub

Private Sub AnnualPayment_Click()
frmLoanPaymentAnnualPayment.Show
End Sub

Private Sub cmdHelp_Click()
frmLoanPaymentHelp.Show
End Sub

Private Sub cmdMenu_Click()
Unload frmLoanPayment
frmFinancialDecisionSupport.Show
End Sub

Private Sub cmdOK_Click()
If IsNumeric(txtFutureValue.Text) And IsNumeric(txtRatePeriod.Text) And
IsNumeric(txtYearsLoan.Text) And IsNumeric(txtLoanInterest.Text) Then
```

```
Frame1.Visible = True
FutureValue = CDbl(txtFutureValue.Text)
Rate = CDbl(txtRatePeriod.Text)
YearPayment = CDbl(txtYearsLoan.Text)
LoanInterest = CDbl(txtLoanInterest.Text)
Else
  Message$ = "Oouch...!"
  Message$ = Message$ & vbCrLf
  Message$ = Message$ & "Please complete the data in the appropriate blank!"
  MsgBox Message$, vbExclamation + vbOKOnly, "  Incomplete Data Input  "
  txtRatePeriod.SetFocus
End If

Discount = (1 + (Rate / 100)) ^ YearPayment
PresentValue = FutureValue / Discount 'compute PV
txtPresentValue = FormatNumber(PresentValue) 'display PV

NewLoanInterest = 0
For i = 1 To YearPayment
OldLoanInterest = (1 / (1 + (LoanInterest / 100)) ^ i)
NewLoanInterest = OldLoanInterest
Next i

If NewLoanInterest > 0 Then
   Payment = PresentValue * NewLoanInterest
   txtPayments = FormatNumber(Payment) 'compute annual payment
   txtMonthlyPayment = FormatNumber(Payment / 12) 'compute monthly payment
   txtNpayments = (YearPayment * 12) 'compute number of monthly payment
Else
'do nothing
End If

If PresentValue <= 0 And Payment <= 0 Then
  txtPresentValue = vbNullString
  txtPayments = vbNullString
End If
Unload frmFinancialDecisionSupport
End Sub

Private Sub Definition_Click()
frmLoanPaymentDefintion.Show
End Sub

Private Sub Form_Load()
Frame1.Visible = False
End Sub

Private Sub MonthlyPayment_Click()
frmLoanPaymentMonthlyPayment.Show
End Sub

Private Sub NoofYearsLoanMade_Click()
frmLoanPaymentNumberOfYearsLoanmade.Show
End Sub
```

```
Private Sub PVOfTotalLoanPayment_Click()
frmLaonPaymentPresentValueOfTotalPayment.Show
End Sub

Private Sub TotalLoanMade_Click()
frmLoanPaymentTotalLoanMade.Show
End Sub

Private Sub TotalNumberOfMonthlyPayments_Click()
frmLoanPaymentTotalNumberOfMonthlyPayment.Show
End Sub

Private Sub txtFutureValue_Change()
If ((txtFutureValue.Text) <> "") Then
 If IsNumeric(txtFutureValue.Text) Then
  'do nothing
 Else
  Beep
  Message$ = "Oouch...!"
  Message$ = Message$ & vbCrLf
  Message$ = Message & "The total loan payment must be in numeric values (e.g. RM
             2000)."
  MsgBox Message, vbCritical + vbOKOnly, " Wrong Total Loan Payment Values
Entry "
  txtFutureValue.SetFocus
 End If
Else
'do nothing
End If
End Sub
Private Sub txtFutureValue_LostFocus()
If (IsNumeric(txtFutureValue.Text)) Then
 If ((txtFutureValue.Text) > 0) Then
  'do nothing
 Else
  Beep
  Message$ = "Oouch...!"
  Message$ = Message$ & vbCrLf
  Message$ = Message & "The loan made must have value (e.g. RM 15,000)."
  MsgBox Message, vbCritical + vbOKOnly, " Wrong Years Range Entry "
  txtFutureValue.SetFocus
 End If
Else
 'do nothing
End If
End Sub

Private Sub txtLoanInterest_Change()
If IsNumeric(txtLoanInterest.Text) Then
 'do nothing
Else
 Beep
 Message = "Oouch...!"
 Message = Message1$ & vbCrLf
 Message = Message1$ & "The interest rate must be in numeric values (e.g. 11)."
```

```
 MsgBox Message, vbCritical + vbOKOnly, " Wrong Interest Rate Value "
 txtLoanInterest.SetFocus
End If
End Sub
Private Sub txtLoanInterest_LostFocus()
If ((txtLoanInterest.Text) <> "") Then
 'txtLoanInterest.Text = ((txtLoanInterest) / 100#)
 If (((txtLoanInterest.Text) >= 0) And ((txtLoanInterest.Text) <= 100)) Then
  'do nothing
 Else
  Beep
  Message$ = "Hmmm...!"
  Message$ = Message$ & vbCrLf
  Message$ = Message$ & "I'm sorry that your request is beyond my forecasting
             ability!"
  MsgBox Message$, vbCritical + vbOKOnly, " Wrong Required Interest Rate Range "
  txtLoanInterest.SetFocus
 End If
Else
 'do nothing
End If
End Sub
Private Sub txtRatePeriod_LostFocus()
If ((txtRatePeriod.Text) <> "") Then
 'txtRatePeriod.Text = (txtRatePeriod) / 100#
 If (((txtRatePeriod.Text) >= 0) And ((txtRatePeriod.Text) <= 100)) Then
  'do nothing
 Else
  Beep
  Message$ = "Hmmm...!"
  Message$ = Message$ & vbCrLf
  Message$ = Message$ & "I'm sorry that your request is beyond my forecasting ablity!
"
  MsgBox Message$, vbCritical + vbOKOnly, " Wrong Required Interest Rate Range "
  txtRatePeriod.SetFocus
 End If
Else
 'do nothing
End If
End Sub

Private Sub txtRatePeriod_Change()
If IsNumeric(txtRatePeriod.Text) Then
 'do nothing
Else
 Beep
 Message = "Oouch...!"
 Message = Message1$ & vbCrLf
 Message = Message1$ & "The interest rate must be in numeric values (e.g. 11)."
 MsgBox Message, vbCritical + vbOKOnly, " Wrong Interest Rate Value "
 txtRatePeriod.SetFocus
End If
End Sub

Private Sub txtYearsLoan_Change()
```

```
If ((txtYearsLoan.Text) <> "") Then
 If IsNumeric(txtYearsLoan.Text) Then
   'do nothing
 Else
   Beep
   Message$ = "Oouch...!"
   Message$ = Message$ & vbCrLf
   Message$ = Message & "The total years must be in numeric values (e.g. 20)."
   MsgBox Message, vbCritical + vbOKOnly, " Wrong Total Years Values Entry "
   txtYearsLoan.SetFocus
 End If
Else
 'do nothing
End If
End Sub
**********************************************************************

frmOperatingBudgeting1

Option Explicit
Private Sub cmdCancel_Click()
txtEquipmentPayment = vbNullString
txtInsurance = vbNullString
txtLaborCostUnit = vbNullString
txtMaterialCostUnit = vbNullString
txtPayrollExpenses = vbNullString
txtPriceUnit = vbNullString
txtRent = vbNullString
txtTransportingCostUnit = vbNullString
txtUnitSales = vbNullString
End Sub

Private Sub cmdHelp_Click()
frmOperatingBudgetingHelp.Show
End Sub
Private Sub cmdMenu_Click()
Unload frmOperatingBudgeting1
frmFinancialDecisionSupport.Show
End Sub

Private Sub cmdOK_Click()
On Error Resume Next

Err.Clear
If Not (IsNumeric(txtEquipmentPayment) And IsNumeric(txtRent) And
  IsNumeric(txtUnitSales) And IsNumeric(txtTransportingCostUnit) And
  IsNumeric(txtTransportingCostUnit) And IsNumeric(txtInsurance) And
  IsNumeric(txtLaborCostUnit) And IsNumeric(txtMaterialCostUnit) And
  IsNumeric(txtPayrollExpenses) And IsNumeric(txtPriceUnit)) Then

  Message1$ = "Oouch...!"
  Message1$ = Message1$ & vbCrLf
  Message1$ = Message1$ & " You haven't complete the details."
  MsgBox Message1$, vbExclamation + vbOKOnly, " Incompelete Details "
Else
```

```
Set obExcelApp = GetObject(, "Excel.Application")
  If Err.Number <> 0 Then
    Err.Clear
    Set obExcelApp = CreateObject("Excel.Application")
    If Err.Number <> 0 Then
      MsgBox "Error: " & Err.Description
    Else
      MyExcel = True
    End If
  Else
    MyExcel = False
  End If
  With obExcelApp
    Set obWorkbooks = GetObject("C:\Project\Budgeting1.xls")
    With obWorkbooks
      Set obWorkSheet = obWorkbooks.Worksheets("Budgeting")
        obWorkSheet.Range("$D$3") = txtYear
        obWorkSheet.Range("$D$5") = txtUnitSales
        obWorkSheet.Range("$D$6") = txtPriceUnit
        obWorkSheet.Range("$D$8") = txtMaterialCostUnit
        obWorkSheet.Range("$D$9") = txtLaborCostUnit
        obWorkSheet.Range("$D$10") = txtTransportingCostUnit
        obWorkSheet.Range("$D$12") = txtRent
        obWorkSheet.Range("$D$13") = txtPayrollExpenses
        obWorkSheet.Range("$D$14") = txtInsurance
        obWorkSheet.Range("$D$15") = txtEquipmentPayment
      End With
      obWorkbooks.Save = True
      obWorkbooks.Close = True
      obWorkbooks.Quit = True
    End With
    obExcelApp.Visible = False
    obExcelApp.Save = True
    obExcelApp.Close = True
    frmOperatingBudgeting2.Show
  End If
  Unload frmFinancialDecisionSupport
  End Sub

  Private Sub Definition_Click()
  frmOperatingBudgetingDefinition.Show
  End Sub

  Private Sub EquipmentPayment_Click()
  frmOperatingBudgetingEquipmentPayment.Show
  End Sub

  Private Sub Insurance_Click()
  frmOperatingBudgetingInsurance.Show
  End Sub

  Private Sub LaborCostPerUnit_Click()
  frmOperatingBudgetingLaborCost.Show
  End Sub
```

```
Private Sub MaterialCostPerUnit_Click()
frmOperatingBudgetingMaterialCost.Show
End Sub

Private Sub PayrollExpenses_Click()
frmOperatingBudgetingPayrollExpenses.Show
End Sub

Private Sub PricePerUnit_Click()
frmOperatingBudgetingPricePerUnit.Show
End Sub

Private Sub Rent_Click()
frmOperatingBudgetingRent.Show
End Sub

Private Sub TransportaionCostPerUnit_Click()
frmOperatingBudgetingTransportationCost.Show
End Sub

Private Sub txtEquipmentPayment_Change()
If ((txtEquipmentPayment.Text) <> "") Then
 If IsNumeric(txtEquipmentPayment) Then
  'do nothing
 Else
  txtEquipmentPayment.Text = vbNullString
  Beep
  Message1$ = "Oouch...!"
  Message1$ = Message1$ & vbCrLf
  Message1$ = Message1$ & "The equipment payment must be in numeric values (e.g.
              RM250,000)."
  MsgBox Message1$, vbCritical + vbOKOnly, "Wrong Equipment Payment Per Unit
Entry"
  txtEquipmentPayment.SetFocus
 End If
Else
 'do nothing
End If
End Sub
Private Sub txtInsurance_Change()
If ((txtInsurance.Text) <> "") Then
 If IsNumeric(txtInsurance) Then
  'do nothing
 Else
  txtInsurance.Text = vbNullString
  Beep
  Message1$ = "Oouch...!"
  Message1$ = Message1$ & vbCrLf
  Message1$ = Message1$ & "The insurance payment must be in numeric values (e.g.
              RM250,000)."
  MsgBox Message1$, vbCritical + vbOKOnly, "Wrong Insurance Payment Per Unit
Entry"
  txtInsurance.SetFocus
 End If
Else
```

```
 'do nothing
End If
End Sub
Private Sub txtLaborCostUnit_Change()
If ((txtLaborCostUnit.Text) <> "") Then
 If IsNumeric(txtLaborCostUnit) Then
  'do nothing
 Else
  txtLaborCostUnit.Text = vbNullString
  Beep
  Message1$ = "Oouch...!"
  Message1$ = Message1$ & vbCrLf
  Message1$ = Message1$ & "The labor cost per unit must be in numeric values (e.g.
              RM2.50)."
  MsgBox Message1$, vbCritical + vbOKOnly, "Wrong Labor Cost Per Unit Entry"
  txtLaborCostUnit.SetFocus
 End If
Else
 'do nothing
End If
End Sub
Private Sub txtMaterialCostUnit_Change()
If ((txtMaterialCostUnit.Text) <> "") Then
 If IsNumeric(txtMaterialCostUnit) Then
  'do nothing
 Else
  txtMaterialCostUnit.Text = vbNullString
  Beep
  Message1$ = "Oouch...!"
  Message1$ = Message1$ & vbCrLf
  Message1$ = Message1$ & "The material cost per unit must be in numeric values
(e.g.
              RM2.50)."
  MsgBox Message1$, vbCritical + vbOKOnly, "Wrong Material Cost Per Unit Entry"
  txtMaterialCostUnit.SetFocus
 End If
Else
 'don nothing
End If
End Sub

Private Sub txtPayrollExpenses_Change()
If ((txtPayrollExpenses.Text) <> "") Then
 If IsNumeric(txtPayrollExpenses) Then
  'do nothing
 Else
  txtPayrollExpenses.Text = vbNullString
  Beep
  Message1$ = "Oouch...!"
  Message1$ = Message1$ & vbCrLf
  Message1$ = Message1$ & "The payroll expenses must be in numeric values (e.g.
              RM250,000)."
  MsgBox Message1$, vbCritical + vbOKOnly, "Wrong Payroll Expenses Entry"
  txtPayrollExpenses.SetFocus
 End If
```

```
Else
 'do nothing
End If
End Sub
Private Sub txtPriceUnit_Change()
If ((txtPriceUnit.Text) <> "") Then
 If IsNumeric(txtPriceUnit) Then
  'do nothing
 Else
  txtPriceUnit.Text = vbNullString
  Beep
  Message1$ = "Oouch...!"
  Message1$ = Message1$ & vbCrLf
  Message1$ = Message1$ & "The price per unit must be in numeric values (e.g.
RM2.50)."
  MsgBox Message1$, vbCritical + vbOKOnly, "Wrong Price Per Unit Entry"
  txtPriceUnit.SetFocus
 End If
Else
 'do nothing
End If
End Sub
Private Sub txtRent_Change()
If ((txtRent.Text) <> "") Then
 If IsNumeric(txtRent) Then
  'do nothing
 Else
  txtRent.Text = vbNullString
  Beep
  Message1$ = "Oouch...!"
  Message1$ = Message1$ & vbCrLf
  Message1$ = Message1$ & "The rental must be in numeric values (e.g.
RM250,000)."
  MsgBox Message1$, vbCritical + vbOKOnly, "Wrong Rental Entry"
  txtRent.SetFocus
 End If
Else
 'do nothing
End If
End Sub
Private Sub txtTransportingCostUnit_Change()
If ((txtTransportingCostUnit.Text) <> "") Then
 If IsNumeric(txtTransportingCostUnit) Then
  FormatNumber (txtTransportingCostUnit)
 Else
  txtTransportingCostUnit.Text = vbNullString
  Beep
  Message1$ = "Oouch...!"
  Message1$ = Message1$ & vbCrLf
  Message1$ = Message1$ & "The transporting cost per unit must be in numeric values
(e.g. RM250,000)."
  MsgBox Message1$, vbCritical + vbOKOnly, "Wrong Transporting Cost Per Unit
Entry"
  txtTransportingCostUnit.SetFocus
 End If
```

```
Else
 'do nothing
End If
End Sub
Private Sub txtUnitSales_Change()
If ((txtUnitSales.Text) <> "") Then
 If IsNumeric(txtUnitSales) Then
  ' do nothing
 Else
  txtUnitSales.Text = vbNullString
  Beep
  Message1$ = "Oouch...!"
  Message1$ = Message1$ & vbCrLf
  Message1$ = Message1$ & "The sales unit must be in numeric values (e.g.
RM250,000)."
  MsgBox Message1$, vbCritical + vbOKOnly, "Wrong Unit Sales Entry"
  txtUnitSales.SetFocus
 End If
Else
 'do nothing
End If
End Sub
Private Sub txtYear_Change()
If ((txtYear.Text) <> "") Then
 If IsNumeric(txtYear) Then
  ' do nothing
 Else
  txtYear.Text = vbNullString
  Beep
  Message1$ = "Oouch...!"
  Message1$ = Message1$ & vbCrLf
  Message1$ = Message1$ & "The year value must be in numeric values (e.g. 1999)."
  MsgBox Message1$, vbCritical + vbOKOnly, "Wrong Year Value Entry"
  txtYear.SetFocus
  End If
 Else
  'do nothing
 End If
 End Sub
 Private Sub txtYear_LostFocus()
 If IsNumeric(txtYear) Then
  If ((txtYear.Text) >= 1990) Then
   'do nothing
  Else
   txtYear.Text = vbNullString
   Beep
   Message1$ = "Hmm...!"
   Message1$ = Message1$ & vbCrLf
   Message1$ = Message1$ & "The year range must be more than 1990 (e.g. 1999)."
   MsgBox Message1$, vbCritical + vbOKOnly, "Wrong Year Range Entry"
   txtYear.SetFocus
   End If
  Else
   'do nothing
  End If
```

```
End Sub
**************************************************************************

frmOperatingBudgeting2

Option Explicit
Private Sub cmdCancel_Click()
Unload frmOperatingBudgeting2
End Sub

Private Sub cmdOK_Click()
If (optForecasting = False) And (optGoalSeeking = False) Then
   Message$ = "Oouch..... !"
   Message$ = Message$ & vbCrLf
   Message$ = Message$ & "You haven't choose your choice! "
   Message$ = Message$ & vbCrLf
   MsgBox Message$, vbExclamation + vbOKOnly, " Invalid Choice Entered "
Else
  If (optGoalSeeking = False) Then
  frmOperatingBudgeting3a.Show
  Else
   frmGoalSeeking.Show
  End If
End If
End Sub
**************************************************************************
frmOperatingBudgeting3a

Option Explicit
Private Sub cmdOK_Click()
Combo1.List(0) = "Report"
Combo1.List(1) = "Graph"

If Combo1.ListIndex = -1 Then
 Message$ = "Oouch..... !"
 Message$ = Message$ & vbCrLf
 Message$ = Message$ & "You haven't choose your choice! "
 Message$ = Message$ & vbCrLf
 MsgBox Message$, vbExclamation + vbOKOnly, " Invalid Choice Entered "
 Combo1.SetFocus
Else
 If Combo1.ListIndex = 0 Then
  frmOperatingBudgetReport.Show
 Else
  If Combo1.ListIndex = 1 Then
   frmOperatingBudgetingGraph.Show
  Else
   'do nothing
  End If
 End If
End If

End Sub

Private Sub cmdCancel_Click()
```

```
Unload Me
End Sub
**************************************************************************

frmOperatingBudgetingGraph

Option Explicit
Private Sub cmdCancel_Click()
Unload frmOperatingBudgetingGraph
End Sub
Private Sub cmdHelp_Click()
frmOperatingBudgetingGraphHelp.Show
End Sub
Private Sub cmdOK_Click()
On Error Resume Next
Err.Clear
   Set obExcelApp = GetObject(, "Excel.Application")
   If Err.Number <> 0 Then
   Err.Clear
      Set obExcelApp = CreateObject("Excel.Application")
      If Err.Number <> 0 Then
        MsgBox "Error: " & Err.Description
      Else
        MyExcel = True
      End If
   Else
      MyExcel = False
   End If

If (Combo1.ListIndex = -1) Then
  Beep
  Message$ = "Oouch...!"
  Message$ = Message$ & vbCrLf
  Message$ = Message$ & "You haven't enter your choice! "
  MsgBox Message$, vbExclamation + vbOKOnly, " Invalid Choice Enter "
  Combo1.SetFocus
Else
  If (Combo1.ListIndex = 0) Then
    With obExcelApp
      Set obWorkbooks = GetObject("C:\Project\Budgeting1.xls")
      With obWorkbooks
        Set obWorkSheet = obWorkbooks.Worksheets("Budgeting")
          obWorkSheet.Range("$I$5").Value = txtRate
          obWorkSheet.Range("$I$6").Value = txtUnitSalesChanges

      Set ChartData = obWorkbooks.Worksheets("ChartData")
         ChartData.Range("B2:B7").NumberFormat = "0"
         ChartData.Range("C2:C7").NumberFormat = "0.00"

         Set ChartSheet = obWorkbooks.Charts.Add
            ChartSheet.ChartType = xlLineMarkers
            ChartSheet.SetSourceData ChartData.Range("A1:C7"), xlColumns
            'ChartSheet.SeriesCollection(1).XValues = "=ChartData!R2C1:R7C1"
            'ChartSheet.SeriesCollection(1).Values = "=ChartData!R2C2:R7C2"
            ChartSheet.SeriesCollection(2).XValues = "=ChartData!R2C1:R7C1"
```

```
            ChartSheet.SeriesCollection(2).Values = "=ChartData!R2C3:R7C3"
            ChartSheet.SeriesCollection(1).AxisGroup = 2
            ChartSheet.HasTitle = True
            ChartSheet.ChartTitle.Characters.Text = _
            "Break-Even Sales vs Margin of Safety"
            ChartSheet.Axes(xlCategory, xlPrimary).HasTitle = True
            ChartSheet.Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = _
            "Year"
            ChartSheet.Axes(xlValue, xlPrimary).HasTitle = True
            ChartSheet.Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = _
            "Margin of Safety (%)"
            ChartSheet.Axes(xlValue, xlSecondary).HasTitle = True
            ChartSheet.Axes(xlValue, xlSecondary).AxisTitle.Characters.Text = _
            "Break-Even Sales (units)"
            ChartSheet.SeriesCollection(1).Delete
            ChartSheet.ChartArea.Copy
            Image1.Picture = Clipboard.GetData(vbCFBitmap)
    End With
     obWorkbooks.Save = True
     obWorkbooks.Close = True
     obWorkbooks.Quit = True
  End With
   obExcelApp.Visible = False
   obExcelApp.Save
   obExcelApp.Close = True
 Else
  If (Combo1.ListIndex = 1) Then
   With obExcelApp
    Set obWorkbooks = GetObject("C:\Project\Budgeting1.xls")

    With obWorkbooks
     Set obWorkSheet = obWorkbooks.Worksheets("Budgeting")
        obWorkSheet.Range("$I$5").Value = txtRate
        obWorkSheet.Range("$I$6").Value = txtUnitSalesChanges

     Set ChartData = obWorkbooks.Worksheets("ChartData")
       ChartData.Range("B10:B15").NumberFormat = "0"
       ChartData.Range("C10:C15").NumberFormat = "0.00"

     Set ChartSheet = obWorkbooks.Charts.Add
      ChartSheet.ChartType = xlLine
      ChartSheet.SetSourceData ChartData.Range("A9:C15"), xlColumns
      ChartSheet.SeriesCollection(1).Delete
      ChartSheet.SeriesCollection(1).XValues = "=ChartData!R10C1:R15C1"
      ChartSheet.SeriesCollection(2).XValues = "=ChartData!R10C1:R15C1"
     With ChartSheet
      .HasTitle = True
      .ChartTitle.Characters.Text = "Revenue vs Gross Margin"
      .Axes(xlCategory, xlPrimary).HasTitle = True
      .Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = "Year"
      .Axes(xlValue, xlPrimary).HasTitle = True
      .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = _
      "Revenue/Gross Margin"
     End With
     ChartSheet.ChartArea.Copy
```

```
      Image1.Picture = Clipboard.GetData(vbCFBitmap)

     End With
      obWorkbooks.Save = True
      obWorkbooks.Close = True
      obWorkbooks.Quit = True
    End With
     obExcelApp.Visible = False
     obExcelApp.Save
     obExcelApp.Close = True
   End If
  End If
End If
End Sub


Private Sub txtUnitSalesChanges_Change()
If IsNumeric(txtUnitSalesChanges.Text) Then
 'do nothing
Else
 Beep
 Message = "Oouch...!"
 Message = Message1$ & vbCrLf
 Message = Message1$ & "The rate of unit sales changes must be in numeric values
(e.g. 11)."
 MsgBox Message, vbCritical + vbOKOnly, " Wrong Rate Of Unit Sales Changes
Value "
 txtUnitSalesChanges.SetFocus
End If
End Sub

Private Sub txtUnitSalesChanges_LostFocus()
If ((txtUnitSalesChanges.Text) <> "") Then
 txtUnitSalesChanges.Text = (txtUnitSalesChanges) / 100#
 If (((txtUnitSalesChanges) >= 0) And ((txtUnitSalesChanges) <= 100)) Then
  'do nothing
 Else
  Beep
  Message$ = "Hmmm...!"
  Message$ = Message$ & vbCrLf
  Message$ = Message$ & "I'm sorry that your request is beyond my forecasting ablity!
"
  MsgBox Message$, vbCritical + vbOKOnly, " Wrong Rate Of Unit Sales Changes
Range "
  txtUnitSalesChanges.SetFocus
 End If
Else
 'do nothing
End If
End Sub
Private Sub txtRate_Change()
If IsNumeric(txtRate.Text) Then
 'do nothing
Else
 Beep
```

```
 Message = "Oouch...!"
 Message = Message1$ & vbCrLf
 Message = Message1$ & "The inflation rate must be in numeric values (e.g. 11)."
 MsgBox Message, vbCritical + vbOKOnly, " Wrong Inflation Rate Value "
 txtRate.SetFocus
End If
End Sub
Private Sub txtRate_LostFocus()
If ((txtRate.Text) <> "") Then
 txtRate.Text = (txtRate) / 100#
 If (((txtRate) >= 0) And ((txtRate) <= 100)) Then
  'do nothing
 Else
  Beep
  Message$ = "Hmmm...!"
  Message$ = Message$ & vbCrLf
  Message$ = Message$ & "I'm sorry that your request is beyond my forecasting ablity!
"
  MsgBox Message$, vbCritical + vbOKOnly, " Wrong inflation Rate Range "
  txtRate.SetFocus
 End If
Else
 'do nothing
End If
End Sub
**************************************************************************

frmOperatingBudgetReport

Option Explicit
Dim i As Integer
Dim j As Integer

Dim Year0 As Integer
Dim Year1 As Integer
Dim Year2 As Integer
Dim Year3 As Integer
Dim Year4 As Integer
Dim Year5 As Integer

Dim UnitSales0 As Currency
Dim UnitSales1 As Currency
Dim UnitSales2 As Currency
Dim UnitSales3 As Currency
Dim UnitSales4 As Currency
Dim UnitSales5 As Currency

Dim PriceUnit0 As Currency
Dim PriceUnit1 As Currency
Dim PriceUnit2 As Currency
Dim PriceUnit3 As Currency
Dim PriceUnit4 As Currency
Dim PriceUnit5 As Currency

Dim Revenue0 As Currency
```

```
Dim Revenue1 As Currency
Dim Revenue2 As Currency
Dim Revenue3 As Currency
Dim Revenue4 As Currency
Dim Revenue5 As Currency

Dim TotalVariableCostYear0 As Currency
Dim TotalVariableCostYear1 As Currency
Dim TotalVariableCostYear2 As Currency
Dim TotalVariableCostYear3 As Currency
Dim TotalVariableCostYear4 As Currency
Dim TotalVariableCostYear5 As Currency

Dim FixedCostYear0 As Currency
Dim FixedCostYear1 As Currency
Dim FixedCostYear2 As Currency
Dim FixedCostYear3 As Currency
Dim FixedCostYear4 As Currency
Dim FixedCostYear5 As Currency

Dim NetProfitYear0 As Currency
Dim NetProfitYear1 As Currency
Dim NetProfitYear2 As Currency
Dim NetProfitYear3 As Currency
Dim NetProfitYear4 As Currency
Dim NetProfitYear5 As Currency

Private Sub cmdHelp_Click()
frmOperatingBudgetingReportHelp.Show
End Sub

Private Sub cmdOK_Click()
On Error Resume Next
Err.Clear
Set obExcelApp = GetObject(, "Excel.Application")
  If Err.Number <> 0 Then
  Err.Clear
    Set obExcelApp = CreateObject("Excel.Application")
    If Err.Number <> 0 Then
      MsgBox "Error: " & Err.Description
    Else
      MyExcel = True
    End If
  Else
    MyExcel = False
  End If

If Not (IsNumeric(txtInflationRate) And IsNumeric(txtUnitSaleChange)) Then
  Message$ = "Oouch...!"
  Message$ = Message$ & vbCrLf
  Message$ = Message$ & " You haven't finish fill in the details."
  MsgBox Message$, vbExclamation + vbOKOnly, " Incompelete Details "
Else
  With obExcelApp
    Set obWorkbooks = GetObject("C:\Project\Budgeting1.xls")
```

```
With obWorkbooks
 Set obWorkSheet = obWorkbooks.Worksheets("Budgeting")
    obWorkSheet.Range("$I$5") = txtInflationRate
    obWorkSheet.Range("$I$6") = txtUnitSaleChange

    Year0 = obWorkSheet.Range("$D$18")
    Year1 = obWorkSheet.Range("$E$18")
    Year2 = obWorkSheet.Range("$F$18")
    Year3 = obWorkSheet.Range("$G$18")
    Year4 = obWorkSheet.Range("$H$18")
    Year5 = obWorkSheet.Range("$I$18")

    UnitSales0 = obWorkSheet.Range("$D$19")
    UnitSales1 = obWorkSheet.Range("$E$19")
    UnitSales2 = obWorkSheet.Range("$F$19")
    UnitSales3 = obWorkSheet.Range("$G$19")
    UnitSales4 = obWorkSheet.Range("$H$19")
    UnitSales5 = obWorkSheet.Range("$I$19")

    PriceUnit0 = obWorkSheet.Range("$D$20")
    PriceUnit1 = obWorkSheet.Range("$E$20")
    PriceUnit2 = obWorkSheet.Range("$F$20")
    PriceUnit3 = obWorkSheet.Range("$G$20")
    PriceUnit4 = obWorkSheet.Range("$H$20")
    PriceUnit5 = obWorkSheet.Range("$I$20")

    Revenue0 = obWorkSheet.Range("$D$21")
    Revenue1 = obWorkSheet.Range("$E$21")
    Revenue2 = obWorkSheet.Range("$F$21")
    Revenue3 = obWorkSheet.Range("$G$21")
    Revenue4 = obWorkSheet.Range("$H$21")
    Revenue5 = obWorkSheet.Range("$I$21")

    TotalVariableCostYear0 = obWorkSheet.Range("$D$23")
    TotalVariableCostYear1 = obWorkSheet.Range("$E$23")
    TotalVariableCostYear2 = obWorkSheet.Range("$F$23")
    TotalVariableCostYear3 = obWorkSheet.Range("$G$23")
    TotalVariableCostYear4 = obWorkSheet.Range("$H$23")
    TotalVariableCostYear5 = obWorkSheet.Range("$I$23")

    FixedCostYear0 = obWorkSheet.Range("$D$26")
    FixedCostYear1 = obWorkSheet.Range("$E$26")
    FixedCostYear2 = obWorkSheet.Range("$F$26")
    FixedCostYear3 = obWorkSheet.Range("$G$26")
    FixedCostYear4 = obWorkSheet.Range("$H$26")
    FixedCostYear5 = obWorkSheet.Range("$I$26")

    NetProfitYear0 = obWorkSheet.Range("$D$28")
    NetProfitYear1 = obWorkSheet.Range("$E$28")
    NetProfitYear2 = obWorkSheet.Range("$F$28")
    NetProfitYear3 = obWorkSheet.Range("$G$28")
    NetProfitYear4 = obWorkSheet.Range("$H$28")
    NetProfitYear5 = obWorkSheet.Range("$I$28")

 End With
```

```
      obWorkbooks.Save = True
      obWorkbooks.Close = True
      obWorkbooks.Quit = True
   End With
   obExcelApp.Save = True
   obExcelApp.Visible = False
End If

MSFlexGrid1.Cols = 7
MSFlexGrid1.Rows = 8

MSFlexGrid1.FixedCols = 1
MSFlexGrid1.FixedRows = 1

For i = 1 To MSFlexGrid1.Cols - 1
   MSFlexGrid1.Col = i
   For j = 1 To MSFlexGrid1.Rows - 1
      MSFlexGrid1.Row = j
      MSFlexGrid1.CellAlignment = flexAlignRightCenter
      MSFlexGrid1.Text = ""
   Next j
Next i

MSFlexGrid1.Row = 0
For i = 1 To MSFlexGrid1.Cols
   MSFlexGrid1.Cols = i
   MSFlexGrid1.TextMatrix(0, 0) = ""
   MSFlexGrid1.TextMatrix(0, 1) = "    Year"
   MSFlexGrid1.TextMatrix(0, 2) = "    Year"
   MSFlexGrid1.TextMatrix(0, 3) = "    Year"
   MSFlexGrid1.TextMatrix(0, 4) = "    Year"
   MSFlexGrid1.TextMatrix(0, 5) = "    Year"
   MSFlexGrid1.TextMatrix(0, 6) = "    Year"

Next i

MSFlexGrid1.Row = 1
For i = 1 To MSFlexGrid1.Cols
   MSFlexGrid1.TextMatrix(1, 0) = ""
   MSFlexGrid1.TextMatrix(1, 1) = Year0
   MSFlexGrid1.TextMatrix(1, 2) = Year1
   MSFlexGrid1.TextMatrix(1, 3) = Year2
   MSFlexGrid1.TextMatrix(1, 4) = Year3
   MSFlexGrid1.TextMatrix(1, 5) = Year4
   MSFlexGrid1.TextMatrix(1, 6) = Year5
Next i

MSFlexGrid1.Row = 2
For i = 1 To MSFlexGrid1.Cols
   MSFlexGrid1.TextMatrix(2, 0) = "Unit Sales"
   MSFlexGrid1.TextMatrix(2, 1) = Format(UnitSales0, "Currency")
   MSFlexGrid1.TextMatrix(2, 2) = Format(UnitSales1, "Currency")
   MSFlexGrid1.TextMatrix(2, 3) = Format(UnitSales2, "Currency")
   MSFlexGrid1.TextMatrix(2, 4) = Format(UnitSales3, "Currency")
   MSFlexGrid1.TextMatrix(2, 5) = Format(UnitSales4, "currency")
```

```
    MSFlexGrid1.TextMatrix(2, 6) = Format(UnitSales5, "Currency")
Next i

MSFlexGrid1.Row = 3
For i = 1 To MSFlexGrid1.Cols
    MSFlexGrid1.TextMatrix(3, 0) = "Price / Unit"
    MSFlexGrid1.TextMatrix(3, 1) = Format(PriceUnit0, "Currency")
    MSFlexGrid1.TextMatrix(3, 2) = Format(PriceUnit1, "Currency")
    MSFlexGrid1.TextMatrix(3, 3) = Format(PriceUnit2, "Currency")
    MSFlexGrid1.TextMatrix(3, 4) = Format(PriceUnit3, "Currency")
    MSFlexGrid1.TextMatrix(3, 5) = Format(PriceUnit4, "currency")
    MSFlexGrid1.TextMatrix(3, 6) = Format(PriceUnit5, "Currency")
Next i

MSFlexGrid1.Row = 4
For i = 1 To MSFlexGrid1.Cols
    MSFlexGrid1.TextMatrix(4, 0) = "Revenue"
    MSFlexGrid1.TextMatrix(4, 1) = Format(Revenue0, "Currency")
    MSFlexGrid1.TextMatrix(4, 2) = Format(Revenue1, "Currency")
    MSFlexGrid1.TextMatrix(4, 3) = Format(Revenue2, "Currency")
    MSFlexGrid1.TextMatrix(4, 4) = Format(Revenue3, "Currency")
    MSFlexGrid1.TextMatrix(4, 5) = Format(Revenue4, "currency")
    MSFlexGrid1.TextMatrix(4, 6) = Format(Revenue5, "Currency")
Next i

MSFlexGrid1.Row = 5
For i = 1 To MSFlexGrid1.Cols
    MSFlexGrid1.TextMatrix(5, 0) = "Total Variable Costs"
    MSFlexGrid1.TextMatrix(5, 1) = Format(TotalVariableCostYear0, "Currency")
    MSFlexGrid1.TextMatrix(5, 2) = Format(TotalVariableCostYear1, "Currency")
    MSFlexGrid1.TextMatrix(5, 3) = Format(TotalVariableCostYear2, "Currency")
    MSFlexGrid1.TextMatrix(5, 4) = Format(TotalVariableCostYear3, "Currency")
    MSFlexGrid1.TextMatrix(5, 5) = Format(TotalVariableCostYear4, "currency")
    MSFlexGrid1.TextMatrix(5, 6) = Format(TotalVariableCostYear5, "Currency")
Next i

MSFlexGrid1.Row = 6
For i = 1 To MSFlexGrid1.Cols
    MSFlexGrid1.TextMatrix(6, 0) = "Total Fixed Costs"
    MSFlexGrid1.TextMatrix(6, 1) = Format(FixedCostYear0, "Currency")
    MSFlexGrid1.TextMatrix(6, 2) = Format(FixedCostYear1, "Currency")
    MSFlexGrid1.TextMatrix(6, 3) = Format(FixedCostYear2, "Currency")
    MSFlexGrid1.TextMatrix(6, 4) = Format(FixedCostYear3, "Currency")
    MSFlexGrid1.TextMatrix(6, 5) = Format(FixedCostYear4, "currency")
    MSFlexGrid1.TextMatrix(6, 6) = Format(FixedCostYear5, "Currency")
Next i

MSFlexGrid1.Row = 7
For i = 1 To MSFlexGrid1.Cols
    MSFlexGrid1.TextMatrix(7, 0) = "Net Profit"
    MSFlexGrid1.TextMatrix(7, 1) = Format(NetProfitYear0, "Currency")
    MSFlexGrid1.TextMatrix(7, 2) = Format(NetProfitYear1, "Currency")
    MSFlexGrid1.TextMatrix(7, 3) = Format(NetProfitYear2, "Currency")
    MSFlexGrid1.TextMatrix(7, 4) = Format(NetProfitYear3, "Currency")
    MSFlexGrid1.TextMatrix(7, 5) = Format(NetProfitYear4, "currency")
```

```
    MSFlexGrid1.TextMatrix(7, 6) = Format(NetProfitYear5, "Currency")
Next i

End Sub
Private Sub cmdCancel_Click()
Unload Me
End Sub

Private Sub Form_Load()
On Error Resume Next
Err.Clear
Set obExcelApp = GetObject(, "Excel.Application")
   If Err.Number <> 0 Then
   Err.Clear
      Set obExcelApp = CreateObject("Excel.Application")
      If Err.Number <> 0 Then
        MsgBox "Error: " & Err.Description
      Else
        MyExcel = True
      End If
   Else
     MyExcel = False
   End If

   With obExcelApp
     Set obWorkbooks = GetObject("C:\Project\Budgeting1.xls")

     With obWorkbooks
      Set obWorkSheet = obWorkbooks.Worksheets("Budgeting")
         Year0 = obWorkSheet.Range("$D$18")
         UnitSales0 = obWorkSheet.Range("$D$19")
         PriceUnit0 = obWorkSheet.Range("$D$20")
         Revenue0 = obWorkSheet.Range("$D$21")
         TotalVariableCostYear0 = obWorkSheet.Range("$D$23")
         FixedCostYear0 = obWorkSheet.Range("$D$26")
         NetProfitYear0 = obWorkSheet.Range("$D$28")

     End With
     obWorkbooks.Save = True
     obWorkbooks.Close = True
     obWorkbooks.Quit = True
   End With
   obExcelApp.Visible = False
   obExcelApp.Save = True
   obExcelApp.Close = True

MSFlexGrid1.Cols = 7
MSFlexGrid1.Rows = 8

MSFlexGrid1.FixedCols = 1
MSFlexGrid1.FixedRows = 1

For i = 1 To MSFlexGrid1.Cols - 1
   MSFlexGrid1.Col = i
   For j = 1 To MSFlexGrid1.Rows - 1
```

```
      MSFlexGrid1.Row = j
      MSFlexGrid1.CellAlignment = flexAlignRightCenter
      MSFlexGrid1.Text = ""
    Next j
Next i

MSFlexGrid1.Row = 0
For i = 1 To MSFlexGrid1.Cols
   MSFlexGrid1.Cols = i
   MSFlexGrid1.TextMatrix(0, 0) = ""
   MSFlexGrid1.TextMatrix(0, 1) = "Year"
   MSFlexGrid1.CellAlignment = flexAlignRightCenter
Next i

MSFlexGrid1.Row = 1
For i = 1 To MSFlexGrid1.Cols
   MSFlexGrid1.TextMatrix(1, 0) = ""
   MSFlexGrid1.TextMatrix(1, 1) = Year0
Next i

MSFlexGrid1.Row = 2
For i = 1 To MSFlexGrid1.Cols
   MSFlexGrid1.TextMatrix(2, 0) = "Unit Sales"
   MSFlexGrid1.TextMatrix(2, 1) = Format(UnitSales0, "Currency")
Next i

MSFlexGrid1.Row = 3
For i = 1 To MSFlexGrid1.Cols
   MSFlexGrid1.TextMatrix(3, 0) = "Price / Unit"
   MSFlexGrid1.TextMatrix(3, 1) = Format(PriceUnit0, "Currency")
Next i

MSFlexGrid1.Row = 4
For i = 1 To MSFlexGrid1.Cols
   MSFlexGrid1.TextMatrix(4, 0) = "Revenue"
   MSFlexGrid1.TextMatrix(4, 1) = Format(Revenue0, "Currency")
Next i

MSFlexGrid1.Row = 5
For i = 1 To MSFlexGrid1.Cols
   MSFlexGrid1.TextMatrix(5, 0) = "Total Variable Costs"
   MSFlexGrid1.TextMatrix(5, 1) = Format(TotalVariableCostYear0, "Currency")
Next i

MSFlexGrid1.Row = 6
For i = 1 To MSFlexGrid1.Cols
   MSFlexGrid1.TextMatrix(6, 0) = "Total Fixed Costs"
   MSFlexGrid1.TextMatrix(6, 1) = Format(FixedCostYear0, "Currency")
Next i

MSFlexGrid1.Row = 7
For i = 1 To MSFlexGrid1.Cols
   MSFlexGrid1.TextMatrix(7, 0) = "Net Profit"
   MSFlexGrid1.TextMatrix(7, 1) = Format(NetProfitYear0, "Currency")
Next i
```

```
End Sub
Private Sub txtInflationRate_Change()
If IsNumeric(txtInflationRate.Text) Then
 'do nothing
Else
 Beep
 Message = "Oouch...!"
 Message = Message1$ & vbCrLf
 Message = Message1$ & "The inflation rate must be in numeric values (e.g. 11)."
 MsgBox Message, vbCritical + vbOKOnly, " Wrong Inflation Rate Value "
 txtInflationRate.SetFocus
End If
End Sub

Private Sub txtInflationRate_LostFocus()
If ((txtInflationRate.Text) <> "") Then
 txtInflationRate.Text = (txtInflationRate) / 100#
 If (((txtInflationRate.Text) >= 0) And ((txtInflationRate.Text) <= 100)) Then
  'do nothing
 Else
  Beep
  Message$ = "Hmmm...!"
  Message$ = Message$ & vbCrLf
  Message$ = Message$ & "I'm sorry that your request is beyond my forecasting ablity!
"
  MsgBox Message$, vbCritical + vbOKOnly, " Wrong Inflation Rate Range "
  txtInflationRate.SetFocus
 End If
Else
 'do nothing
End If
End Sub

Private Sub txtUnitSaleChange_Change()
If IsNumeric(txtUnitSaleChange.Text) Then
 'do nothing
Else
 Beep
 Message = "Oouch...!"
 Message = Message1$ & vbCrLf
 Message = Message1$ & "The rate of unit sales changes must be in numeric values
(e.g. 11)."
 MsgBox Message, vbCritical + vbOKOnly, " Wrong Rate Of Unit Sales Changes
Value "
 txtUnitSaleChange.SetFocus
End If
End Sub

Private Sub txtUnitSaleChange_LostFocus()
If ((txtUnitSaleChange.Text) <> "") Then
 txtUnitSaleChange.Text = (txtUnitSaleChange) / 100#
 If (((txtUnitSaleChange.Text) >= 0) And ((txtUnitSaleChange.Text) <= 100)) Then
  'do nothing
 Else
```

```
   Beep
   Message$ = "Hmmm...!"
   Message$ = Message$ & vbCrLf
   Message$ = Message$ & "I'm sorry that your request is beyond my forecasting ablity!
"
   MsgBox Message$, vbCritical + vbOKOnly, " Wrong Rate Of Unit Sales Changes
Range "
   txtUnitSaleChange.SetFocus
  End If
Else
  'do nothing
End If
End Sub
*************************************************************************
```